COURS D'ANALYSE NUMÉRIQUE

Prof. OKOU Hypolithe

Table des matières

Chapitre 1

Analyse d'erreurs

1.1 Introduction

Les cours traditionnels de mathématiques nous familiarisent avec des théo-ries et des méthodes qui permettent de résoudre de façon analytique un certain nombre de problèmes. C'est le cas notamment des techniques d'intégration et de résolution d'équations algébriques ou différentielles. Bien que l'on puisse proposer plusieurs méthodes pour résoudre un problème donné, celles-ci conduisent à un même résultat, normalement exempt d'erreur.

C'est ici que l'analyse numérique se distingue des autres champs plus classiques des mathématiques. En effet, pour un problème donné, il est possible d'utiliser plusieurs techniques de résolution qui résultent en différents algorithmes. Ces algorithmes dépendent de certains paramètres qui influent sur la précision du résultat. Une partie importante de l'analyse numérique consiste donc à contenir les effets des erreurs ainsi introduites, qui proviennent de trois sources principales :

1. **les erreurs de modélisation ;**
2. **les erreurs de représentation sur ordinateur ;**
3. **les erreurs de troncature.**

1.2 Définitions et Exemples

Définition 1.2.1. *Soit x, un nombre, et x^*, une approximation de ce nombre.*

- *L'erreur absolue est définie par :*
$$\triangle x = |x - x^*|. \tag{1.1}$$
- *L'erreur relative est définie par :*
$$E_r(x^*) = \frac{|x - x^*|}{|x|} = \frac{\triangle x}{|x|} \simeq \frac{\triangle x}{|x^*|}. \tag{1.2}$$

De plus, en multipliant E_r par 100%, on obtient l'erreur relative en pourcentage

Remarque 1.2.1.

- *L'erreur absolue donne une mesure quantitative de l'erreur commise et l'erreur relative en mesure l'importance relativement à la valeur exacte ou la valeur approximative (selon la formule utilisée).*
- *Le problème dans cette définition est qu'on ne connaît pas en général la valeur exacte x de la solution.*
- *Dans certains cas, nous ne connaissons pas la valeur approximative x^*, mais nous disposons souvent d'une borne supérieure pour l'erreur absolue qui dépend de la précision des instruments de mesure utilisés. Cette borne est aussi appelée erreur absolue, alors qu'en fait on a :*

$|x - x^*| \leq \triangle x$ *ce qui est équivalent à :* $x^* - \triangle x \leq x \leq x^* + \triangle x$.
Ainsi, on peut parfois écrire : $x = x^* \pm \triangle x$. *On dira que l'on a estimé la valeur exacte* x *à partir de* x^* *avec une incertitude de* $\triangle x$ *de part et d'autre.*

– *l'erreur* $\triangle x$ *donne la précision avec on a estimé* x. *Elle permet donc de savoir dans l'écriture de* x^* *quels sont les chiffres qui sont significatifs c'est à dire qui sont exacts.*

Définition 1.2.2. *Si l'erreur absolue vérifie :*

$$\triangle x \leq 0,5 \times 10^m \tag{1.3}$$

alors le chiffre correspondant à la m^{ieme} *puissance de* 10 *est dit significatif et tous les autres chiffres à sa gauche, correspondant aux puissances de* 10 *supérieures à* m, *le sont aussi. On arrête le compte au dernier chiffre non nul.*

Remarque 1.2.2.
Il existe une exception à la règle. Si le chiffre correspondant à la m^{ieme} *puissance de* 10 *est nul ainsi que tous ceux à sa gauche, on dit qu'il n'y a aucun chiffre significatif. Inversement, si un nombre est donné avec* n *chiffres significatifs, on commence à compter à partir du premier chiffre non nul à gauche.*

Remarque 1.2.3.
Nous cherchons à minimiser l'erreur $\triangle x$, *donc nous cherchons à déterminer la valeur de* m *la plus petite possible.*

Exemple 1.2.1.

1. *On obtient une approximation de* $x = \pi$ *au moyen de la quantité* $x^* = \dfrac{22}{7} = 3,142857....$ *On en conclut que :* $\triangle x = |\pi - \dfrac{22}{7}| = 0,00126.... \simeq 0,126 \times 10^{-2}$. *Puisque l'erreur absolue est plus petite que* $0,5 \times 10^{-2}$, *le chiffre des centièmes est significatif et on a en tout* 3 *chiffres significatifs qui sont* 3 ; 1 ; 4.
2. *Supposons que* $x = e$ *et* $x^* = 2,71824537$.
   On a $\triangle x = |e - 2,71824537| = 0,000036458... = 0,36458... \times 10^{-4} \leq 0,5 \times 10^{-4}$. *Cela veut dire que le* 4^{ieme} *chiffre après la virgule, (2) est signaficatif. Ainsi, on a* $e \simeq 2,7182 = x^{**}$ *et tous les chiffres de* x^{**} *sont significatifs. Par suite l'erreur commise pour cette approximation vérifie* $\triangle x' \leq 0,5 \times 10^{-4}$.

1.3 Erreur de modélisation

Comme leur nom l'indique, proviennent de l'étape de mathématisation du phénomène physique auquel on s'intéresse. Cette étape consiste à faire ressortir les causes les plus déterminantes du phénomène observé et à les mettre sous forme d'équations (différentielles le plus souvent). Si le phénomène observé est très complexe, il faut simplifier et négliger ses composantes qui paraissent moins importantes ou qui rendent la résolution numérique trop difficile. C'est ce que l'on appelle les erreurs de modélisation.

L'effort de modélisation produit en général des systèmes d'équations complexes qui comprennent un grand nombre de variables inconnues. Pour réussir à les résoudre, il faut simplifier certaines composantes et négliger les moins importantes. On fait alors une première erreur de modélisation. De plus, même bien décomposé, un phénomène physique peut être difficile à mettre sous forme d'équations. On introduit alors un modèle qui décrit au mieux son influence, mais qui demeure une approximation de la réalité. On commet alors une deuxième erreur de modélisation.

1.4 Erreur de représentation sur ordinateur

Elles sont liées à l'utilisation de l'ordinateur. En effet, la représentation sur ordinateur (généralement binaire) des nombres introduit souvent des erreurs. Même infimes au départ, ces erreurs peuvent s'accumuler lorsqu'on effectue un très grand nombre d'opérations. Ces erreurs se propagent au fil des calculs et peuvent compromettre la précision des résultats.

1.5 Erreurs de troncature

Elles proviennent principalement de l'utilisation du développement de Taylor, qui permet par exemple de remplacer une équation différentielle par une équation algébrique. Le développement de Taylor est le principal outil mathématique du numéricien. Il est primordial d'en maîtriser l'énoncé et ses conséquences.

Les erreurs de troncature constituent la principale catégorie d'erreurs. Elles sont liées aux méthodes de résolution utilisées et qui comportent la plupart de temps des erreurs de troncature plus ou moins importantes. L'ordre d'une méthode dépend du nombre de termes utilisés dans les développements de Taylor appropriés. Il est donc essentiel de revoir en détail le développement de Taylor, car il constitue l'outil fondamental de l'analyse numérique.

Chapitre 2

Systèmes d'équations algébriques

Les systèmes d'équations algébriques jouent un rôle très important en in-génierie. On peut classer ces systèmes en deux grandes familles : les systèmes linéaires et les systèmes non linéaires. Ici encore, les progrès de l'informatique et de l'analyse numérique per-mettent d'aborder des problèmes de taille prodigieuse. On résout couramment aujourd'hui des systèmes de plusieurs centaines de milliers d'inconnues. On rencontre ces applications en mécanique des fluides et dans l'analyse de struc-tures complexes. On peut par exemple calculer l'écoulement de l'air autour d'un avion ou l'écoulement de l'eau dans une turbine hydraulique complète. On peut également analyser la résistance de la carlingue d'un avion à différentes contraintes extérieures et en vérifier numériquement la solidité.

2.1 Systèmes d'équations linéaires

De façon générale, la résolution d'un système d'équations linéaires consiste à trouver un vecteur $\overrightarrow{x} = [x_1\ x_2\ x_3\ ...\ x_n]^t$ ($\overrightarrow{x}$ dénotera toujours un vecteur colonne et l'indice supérieur t symbolisera sa transposée) solution de :

$$\left\{\begin{array}{lllll} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + ... + a_{1n}x_n & = & b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + ... + a_{2n}x_n & = & b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + ... + a_{3n}x_n & = & b_3 \\ ... \quad ... \quad ... \quad ... & ... & ... \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + ... + a_{nn}x_n & = & b_n \end{array}\right. \tag{2.1}$$

On peut utiliser la notation matricielle, qui est beaucoup plus pratique et surtout plus compacte. On écrit alors le système précédent sous la forme :

$$Ax = b \tag{2.2}$$

où A est la matrice :

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & ... & a_{1n} \\ a_{21} & a_{22} & a_{23} & ... & a_{2n} \\ a_{31} & a_{32} & a_{33} & ... & a_{3n} \\ ... & ... & ... & ... & ... \\ a_{n1} & a_{n2} & a_{n3} & ... & a_{nn} \end{pmatrix}$$

et où $b = [b_1\ b_2\ b_3\ ....\ b_n]^t$ est le membre de droite. Bien entendu, la matrice A et le vecteur b sont connus. Il reste à déterminer le vecteur x. Le problème 2.1 (ou 2.2) est un système de n équations et n inconnues. En pratique, la valeur de n varie considérablement et peut s'élever jusqu'à plusieurs centaines de milliers. Dans ce chapitre, nous nous limitons à des systèmes de petite taille, mais les stratégies développées sont valides pour des systèmes de très grande taille. Notons toutefois que le coût de la résolution croît rapidement avec n.

Remarque 2.1.1. *Dans la plupart des cas, nous traitons des matrices non singulières ou inversibles, c'est-à-dire dont la matrice inverse existe. Nous ne faisons pas non plus de révision systématique de l'algèbre linéaire élémentaire que nous supposons connue. Ainsi, la solution de l'équation 2.2) peut s'écrire : $x = A^{-1}b$ et la discussion peut sembler close. Nous verrons cependant que le calcul de la matrice inverse A^{-1} est plus difficile et plus long que la résolution du système linéaire de départ.*

Définition 2.1.1. *Une matrice est dite triangulaire inférieure (ou supérieure) si tous les a_{ij} (ou tous les a_{ji}) sont nuls pour $i < j$. Une matrice triangulaire inférieure a la forme type :*

$$\begin{pmatrix} a_{11} & 0 & 0 & ... & 0 \\ a_{21} & a_{22} & 0 & ... & 0 \\ ... & ... & ... & ... & ... \\ a_{31} & a_{32} & ... & a_{(n-1)(n-1)} & 0 \\ a_{n1} & a_{n2} & ... & a_{n(n-1)} & a_{nn} \end{pmatrix}$$

Une matrice triangulaire supérieure est tout simplement la transposée d'une matrice triangulaire inférieure.

Les systèmes triangulaires sont également faciles à résoudre. Il suffit en effet de commencer par l'équation qui se trouve à la pointe du triangle (la pre-mière pour une matrice triangulaire inférieure et la dernière pour une matrice triangulaire supérieure) et de résoudre une à une les équations. On parle de descente triangulaire ou de remontée triangulaire, selon le cas.

Définition 2.1.2. *Une méthode de résolution d'un système linéaire est dite directe si la solu-tion du système peut être obtenue par cette méthode en un nombre fini et prédéterminé d'opérations.*

Autrement dit, les méthodes directes permettent d'obtenir le résultat après un nombre connu de multiplications, divisions, additions et soustractions. On peut alors en déduire le temps de calcul nécessaire à la résolution (qui peut être très long si n est grand). Les méthodes directes s'opposent sur ce point aux méthodes dites itératives, qui peuvent converger en quelques itérations, converger en un très grand nombre d'itérations ou même diverger, selon le cas.

2.2 Opérations élémentaires sur les lignes

Revenons au système :

$$Ax = b \tag{2.3}$$

et voyons comment on peut le transformer sans en modifier la solution. La réponse est toute simple. On peut toujours multiplier (à gauche de chaque côté) les termes de cette relation par une matrice W inversible ; la solution n'est pas modifiée puisque l'on peut remultiplier par W^{-1} pour revenir au système de départ. Ainsi : $WAx = Wb$ possède la même solution que le système 2.3.

Pour transformer un système quelconque en système triangulaire, il suffit d'utiliser trois opérations élémentaires sur les lignes de la matrice. Ces trois opérations élémentaires correspondent à trois types différents de matrices W. C'est la base de la méthode d'élimination de Gauss. On note l_i, la ligne i de la matrice A. Cette notation est quelque peu ambiguë, car on se trouve de ce fait à placer une ligne de la matrice A dans un vecteur colonne. Cela n'empêche cependant pas la compréhension de la suite. Les trois opérations élémentaires dont on a besoin sont les suivantes :

1. opération ($l_i \longleftarrow l_i$) : remplacer la ligne i par un multiple d'elle-même ;
2. opération ($l_i \longleftrightarrow l_j$) : intervertir la ligne i et la ligne j ;
3. opération ($l_i \longleftarrow l_i + \lambda l_j$) : remplacer la ligne i par la ligne i plus un multiple de la ligne j.

Ces trois opérations élémentaires sont permises, car elles équivalent à multiplier le système 2.1 par une matrice inversible.

2.2.1 Multiplication d'une ligne par un scalaire

Remplacer la ligne i par un multiple d'elle-même ($l_i \longleftarrow \lambda l_i$) revient à multi-plier le système linéaire 2.2 par une matrice diagonale inversible $W = M(l_i \longleftarrow \lambda l_i)$, dont tous les éléments diagonaux sont 1, sauf m_{ii}, qui vaut λ. Tous les autres termes sont nuls. Cette matrice a pour effet de multiplier la ligne i par le scalaire λ.

Remarque 2.2.1. *Le déterminant de la matrice diagonale $M(l_i \longleftarrow \lambda l_i)$ est λ. La matrice est donc inversible si $\lambda \neq 0$.*

Remarque 2.2.2. *La matrice inverse de $M(l_i \longleftarrow \lambda l_i)$ est simplement $M(l_i \longleftarrow \lambda^{-1} l_i)$, c'est-à-dire :*

$$M^{-1}(l_i \longleftarrow \lambda^{-1} l_i) = M(l_i \longleftarrow \lambda^{-1} l_i) \tag{2.4}$$

Il suffit donc de remplacer λ par $\dfrac{1}{\lambda}$ pour inverser la matrice.

Exemple 2.2.1. *Soit le système :*

$$\begin{pmatrix} 3 & 1 & 2 \\ 6 & 4 & 1 \\ 5 & 4 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 11 \\ 10 \end{pmatrix} \tag{2.5}$$

dont la solution est $x = [1 \;\; 1 \;\; 1]^t$. Si l'on souhaite multiplier la ligne 2 par un facteur 3, cela revient à multiplier le système par la matrice :

$$M(l_2 \longleftarrow 3l_2) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

et l'on obtient :

$$\begin{pmatrix} 3 & 1 & 2 \\ 18 & 12 & 3 \\ 5 & 4 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 33 \\ 10 \end{pmatrix}.$$

La solution de ce nouveau système reste la même que celle du système de départ puisque la matrice $M(l_2 \longleftarrow 3l_2)$ est inversible (et son déterminant est 3).

2.2.2 Permutation de deux lignes

L'opération élémentaire qui consiste à intervertir deux lignes ($l_i \longleftrightarrow l_j$) est également connue sous le nom de permutation de lignes. Cette opération est équivalente à la multiplication du système 2.2 par une matrice inversible $W = P(l_i \longleftarrow l_j)$, obtenue en permutant les lignes i et j de la matrice identité.

Exemple 2.2.2. *On veut intervertir la ligne 2 et la ligne 3 du système de l'exemple précédent. Il suffit de le multiplier par la matrice :*

$$P(l_2 \longleftrightarrow l_3) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

et l'on obtient :

$$\begin{pmatrix} 3 & 1 & 2 \\ 5 & 4 & 1 \\ 6 & 4 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 10 \\ 11 \end{pmatrix}.$$

La matrice $P(l_i \longleftrightarrow l_j)$ est inversible. Pour obtenir son inverse, il suffit de réfléchir une seconde. En effet, quelle est l'opération inverse de celle qui inverse deux lignes, sinon l'inversion des deux mêmes lignes ?

Remarque 2.2.3. *L'inverse de la matrice $P(l_i \longleftrightarrow l_j)$ est donc la matrice $P(l_i \longleftrightarrow l_j)$ elle-même, c'est-à-dire :*

$$P^{-1}(l_i \longleftrightarrow l_j) = P(l_i \longleftrightarrow l_j).$$

Remarque 2.2.4. *On montre assez facilement que le déterminant de $P(l_i \longleftrightarrow l_j)$ est -1. Lorsque l'on permute deux lignes, le déterminant du système de départ change de signe.*

2.2.3 Opération $(l_i \longleftarrow l_i + \lambda l_j)$

La dernière opération élémentaire consiste à remplacer la ligne i par la ligne i plus un multiple de la ligne j $(l_i \longleftarrow l_i + \lambda l_j)$. Cela est encore une fois équivalent à multiplier le système de départ par une matrice inversible $W = T(l_i \longleftarrow l_i + \lambda l_j)$ qui vaut 1 sur toute la diagonale et 0 partout ailleurs, sauf t_{ij}, qui vaut λ.

Exemple 2.2.3. *Dans le système 2.5, on souhaite remplacer la deuxième ligne par la deuxième ligne $(i = 2)$ moins deux fois $(\lambda = -2)$ la première ligne $(j = 1)$. Il suffit alors de multiplier le système par :*

$$T(l_2 \longleftarrow l_2 - 2l_1) = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

ce qui donne :

$$\begin{pmatrix} 3 & 1 & 2 \\ 0 & 2 & -3 \\ 5 & 4 & 1 \end{pmatrix} \begin{pmatrix} 6 \\ -1 \\ 10 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}.$$

Remarque 2.2.5. *La matrice $T(l_i \longleftarrow l_i + \lambda l_j)$ est inversible. Pour obtenir son inverse, il suffit de remplacer λ par $-\lambda$, c'est-à-dire :*

$$T^{-1}(l_i \longleftarrow l_i + \lambda l_j) = T(l_i \longleftarrow l_i - \lambda l_j) \tag{2.6}$$

Cela signifie que pour revenir en arrière il suffit de soustraire la ligne que l'on vient d'ajouter.

Remarque 2.2.6. *On peut montrer facilement que le déterminant de $T(l_i \longleftarrow l_i + \lambda l_j)$ est 1.*

Remarque 2.2.7. *Dans cet exemple, en additionnant le bon multiple de la ligne 1 à la ligne 2, on a introduit un 0 à la position a_{21}. En remplaçant la ligne 3 par la ligne 3 moins $\frac{5}{3}$ fois la ligne 1 (ou encore $l_3 \longleftarrow l_3 - \frac{5}{3} l_1$), on introduirait un terme 0 à la position a_{31}. On peut ainsi transformer un système linéaire quelconque en système triangulaire. C'est là la base sur laquelle repose la méthode d'élimination de Gauss.*

Remarque 2.2.8. *Des trois opérations élémentaires, seule l'opération ($l_i \longleftarrow l_i + \lambda l_j$) n'a pas d'effet sur le déterminant. La permutation de deux lignes en change le signe, tandis que la multiplication d'une ligne par un scalaire multiplie le déterminant par ce même scalaire.*

2.3 Élimination de Gauss

Tous les outils sont en place pour la résolution d'un système linéaire. Il suffit maintenant d'utiliser systématiquement les opérations élémentaires pour introduire des zéros sous la diagonale de la matrice A et obtenir ainsi un système triangulaire supérieur. La validité de la méthode d'élimination de Gauss repose sur le fait que les opérations élémentaires consistent à multiplier le système de départ par une matrice inversible.

Remarque 2.3.1. *En pratique, on ne multiplie jamais explicitement les systèmes considérés par les différentes matrices W, car ce serait trop long. Il faut cependant garder en tête que les opérations effectuées sont équivalentes à cette multiplication.*

La méthode d'élimination de Gauss consiste à éliminer tous les termes sous la diagonale de la matrice A. Avant de considérer un exemple, introduisons la matrice augmentée.

Définition 2.3.1. *La matrice augmentée du système linéaire 2.1 est la matrice de dimension n sur $n+1$ que l'on obtient en ajoutant le membre de droite b à la matrice A, c'est-à-dire :*

$$\left(\begin{array}{ccccc|c} a_{11} & a_{12} & a_{13} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} & b_2 \\ \dots & \dots & \dots & \dots & \dots & \\ a_{31} & a_{32} & \dots & a_{(n-1)(n-1)} & a_{(n-1)n} & b_{n-1} \\ a_{n1} & a_{n2} & \dots & a_{n(n-1)} & a_{nn} & b_n \end{array}\right). \tag{2.7}$$

Puisque les opérations élémentaires doivent être effectuées à la fois sur les lignes de la matrice A et sur celles du vecteur b, cette notation est très utile.

Remarque 2.3.2. *Il arrive également que l'on doive résoudre des systèmes de la forme $Ax = b$ avec k seconds membres b différents (la matrice A étant fixée). On peut alors construire la matrice augmentée contenant les k seconds membres désirés. La matrice augmentée ainsi obtenue est de dimension $n \times (n+k)$.*

Exemple 2.3.1. *Considérons l'exemple suivant :*

$$\left(\begin{array}{ccc|c} 2 & 1 & 2 & 10 \\ 6 & 4 & 0 & 26 \\ 8 & 5 & 1 & 35 \end{array}\right) \begin{array}{l} \\ T_1(l_2 \longleftarrow l_2 - (6/2)l_1) \\ T_2(l_3 \longleftarrow l_3 - (8/2)l_1) \end{array}$$

On a indiqué ci-dessus la matrice augmentée de même que les opérations élé-mentaires (et les matrices associées) qui sont nécessaires pour éliminer les termes non nuls sous la diagonale de la première colonne. Il est à noter que l'on divise par $2(a_{11})$ les coefficients qui multiplient la ligne 1. On dit alors que 2 est le pivot. On obtient, en effectuant les opérations indiquées :

$$\left(\begin{array}{ccc|c} 2 & 1 & 2 & 10 \\ 0 & 1 & -6 & -4 \\ 0 & 1 & -7 & -5 \end{array}\right) T_3(l_3 \longleftarrow l_3 - (1/1)l_2)$$

Pour produire une matrice triangulaire supérieure, il suffit maintenant d'intro-duire des 0 sous la diagonale de la deuxième colonne. L'opération est indiquée ci-dessus et le pivot est 1 puisque maintenant $a_{22} = 1$. On obtient donc :

$$\left(\begin{array}{ccc|c} 2 & 1 & 2 & 10 \\ 0 & 1 & -6 & -4 \\ 0 & 0 & -1 & -1 \end{array}\right) \tag{2.8}$$

Il reste ensuite à faire la remontée triangulaire. On obtient :

$$x_3 = \frac{-1}{-1} = 1$$

d'où : $x_2 = \dfrac{-4-(-6)(1)}{1} = 2$ *et enfin :* $x_1 = \dfrac{10-1\times 2-2\times 1}{2} = 3$. *On a construit le système triangulaire 2.8 en effectuant des opérations élémentaires directement sur les lignes de la matrice. La matrice triangulaire obtenue est notée U. Les opérations effectuées pour obtenir U sont équivalentes à multiplier le système de départ par une suite de matrices inversibles. On a en fait :* $U = T_3T_2T_1A$ *où les matrices* T_i *correspondent aux différentes opérations effectuées sur les lignes de la matrice. Plus explicitement, on a :*

$$\begin{pmatrix} 2 & 1 & 2 \\ 0 & 1 & -6 \\ 0 & 0 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -4 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -3 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 2 \\ 6 & 4 & 0 \\ 8 & 5 & 1 \end{pmatrix}.$$

Si l'on poursuit le raisonnement, on a également : $A = T_1^{-1}T_2^{-1}T_3^{-1}U$. *Puisque l'on sait inverser les matrices* T_i *, on a immédiatement que :*

$$\begin{pmatrix} 2 & 1 & 2 \\ 6 & 4 & 0 \\ 8 & 5 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 4 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 2 \\ 0 & 1 & -6 \\ 0 & 0 & -1 \end{pmatrix}$$

ou encore :

$$\begin{pmatrix} 2 & 1 & 2 \\ 6 & 4 & 0 \\ 8 & 5 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 4 & 1 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 2 \\ 0 & 1 & -6 \\ 0 & 0 & -1 \end{pmatrix}.$$

On remarque que les coefficients de la matrice triangulaire inférieure sont ceux qui ont permis d'éliminer les termes non nuls sous la diagonale de la matrice A. Tout cela revient à décomposer la matrice A en un produit d'une matrice triangulaire inférieure, notée L, et d'une matrice triangulaire supérieure U. C'est ce que l'on appelle une décomposition LU.

Exemple 2.3.2. *Soit le système linéaire suivant :*

$$\begin{cases} x_1 + x_2 + 2x_3 + x_4 &= 2 \\ 2x_1 + 2x_2 + 5x_3 + 3x_4 &= 4 \\ x_1 + 3x_2 + 3x_3 + 3x_4 &= -2 \\ x_1 + x_2 + 4x_3 + 5x_4 &= -2 \end{cases}$$

dont la matrice augmentée est :

$$\left(\begin{array}{cccc|c} 1 & 1 & 2 & 1 & 2 \\ 2 & 2 & 5 & 3 & 4 \\ 1 & 3 & 3 & 3 & -2 \\ 1 & 1 & 4 & 5 & -2 \end{array}\right) \begin{array}{l} \\ T_1(l_2 \longleftarrow l_2 - (2/1)l_1) \\ T_2(l_3 \longleftarrow l_3 - (1/1)l_1) \\ T_3(l_4 \longleftarrow l_4 - (1/1)l_1) \end{array}$$

En faisant les opérations indiquées (le pivot a_{11} *est* 1*), on élimine les termes non nuls sous la diagonale de la première colonne et l'on obtient :*

$$\left(\begin{array}{cccc|c} 1 & 1 & 2 & 1 & 2 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 2 & 1 & 2 & -4 \\ 0 & 0 & 2 & 4 & -4 \end{array}\right) P_4(l_2 \longleftrightarrow l_3)$$

Ici, la procédure est interrompue par le fait que le nouveau pivot serait 0 *et qu'il n'est pas possible d'éliminer les termes sous ce pivot. Toutefois, on peut encore, parmi les opérations élémentaires, interchanger deux lignes. Le seul choix possible dans cet exemple est d'intervertir la ligne* 2 *et la ligne* 3*. On se rend immédiatement compte qu'il n'y a plus que des* 0 *sous le nouveau pivot et que l'on peut passer à la colonne suivante :*

$$\left(\begin{array}{cccc|c} 1 & 1 & 2 & 1 & 2 \\ 0 & 2 & 1 & 2 & -4 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 2 & 4 & -4 \end{array}\right) T_5(l_4 \longleftarrow l_4 - (2/1)l_3)$$

En effectuant cette dernière opération, on obtient le système triangulaire :

$$\left(\begin{array}{cccc|c} 1 & 1 & 2 & 1 & 2 \\ 0 & 2 & 1 & 2 & -4 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 2 & -4 \end{array}\right)$$

La remontée triangulaire (laissée en exercice) donne la solution : $x = [1 - 12 - 2]^t$*. Encore ici, la matrice triangulaire est le résultat du produit des opérations élémentaires :*

$$U = T_5 P_4 T_3 T_2 T_1 A$$

ou encore :

$$A = T_1^{-1} T_2^{-1} T_3^{-1} P_4^{-1} T_5^{-1} U$$

qui s'écrit :

$$\begin{pmatrix} 1 & 1 & 2 & 1 \\ 2 & 2 & 5 & 3 \\ 1 & 3 & 3 & 3 \\ 1 & 1 & 4 & 5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 2 & 1 \\ 0 & 2 & 1 & 2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 2 \end{pmatrix}$$

On remarque que la première matrice du terme de droite n'est pas triangulaire inférieure. Cela est dû au fait que l'on a permuté deux lignes. En remultipliant par P_4 des deux côtés la dernière relation, on revient à $P_4 A = LU$ et L est alors triangulaire inférieure.

2.4 Décomposition LU

2.4.1 Principe de la méthode

Supposons un instant que nous ayons réussi à exprimer la matrice A en un produit de deux matrices triangulaires L et U. Comment cela nous permet-il de résoudre le système $Ax = b$? Il suffit de remarquer que :

$$Ax = LUx = b$$

et de poser

$$Ux = y.$$

La résolution du système linéaire se fait alors en deux étapes :

$$\begin{aligned} Ly &= b \\ Ux &= y \end{aligned} \tag{2.9}$$

qui sont deux systèmes triangulaires. On utilise d'abord une descente triangulaire sur la matrice L pour obtenir y et, par la suite, une remontée triangulaire sur la matrice U pour obtenir la solution recherchée x. Il faut tout de suite souligner que la décomposition LU n'est pas unique. On peut en effet écrire un nombre réel comme le produit de deux autres nombres d'une infinité de façons. Il en est de même pour les matrices.

Exemple 2.4.1. *Pour illustrer la non-unicité de la décomposition LU, il suffit de vérifier les égalités :*

$$\begin{pmatrix} 2 & -1 & -1 \\ 0 & -4 & 2 \\ 6 & -3 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & -4 & 0 \\ 6 & 0 & 4 \end{pmatrix} \begin{pmatrix} 1 & -0,5 & -0,5 \\ 0 & 1 & -0,5 \\ 0 & 0 & 1 \end{pmatrix} et : \begin{pmatrix} 2 & -1 & -1 \\ 0 & -4 & 2 \\ 6 & -3 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 3 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & -1 & -1 \\ 0 & -4 & 2 \\ 0 & 0 & 4 \end{pmatrix}.$$

Remarque 2.4.1. *La décomposition LU n'étant pas unique, il faut faire au préalable des choix arbitraires. Le choix le plus populaire consiste à imposer que la matrice U ait des 1 sur sa diagonale. C'est la décomposition de Crout. Certains logiciels comme Matlab [29] préfèrent mettre des 1 sur la diagonale de L. Il en résulte bien sûr une décomposition LU différente de celle de Crout, mais le principe de base reste le même.*

2.4.2 Décomposition de Crout

Algorithme : Décomposition de CROUT

1. Décomposition LU (sans permutation de lignes)
   - Première colonne de L : $l_{i1} = a_{i1}$ pour $i = 1, 2, ..., n$
   - Première ligne de U : $u_{1i} = \dfrac{a_{1i}}{l_{11}}$ pour $i = 2, 3, ..., n$
   - Pour $i = 2, 3, 4, ..., n-1$:
     - Calcul du pivot :

$$l_{ii} = a_{ii} - \sum_{k=1}^{i-1} l_{ik} u_{ki}. \tag{2.10}$$

     - Pour $j = i+1, i+2, ..., n$:
     - Calcul de la i^{eme} colonne de L :

$$l_{ji} = a_{ji} - \sum_{k=1}^{i-1} l_{jk} u_{ki} \tag{2.11}$$

     - Calcul de la i^{eme} ligne de U :

$$u_{ij} = \frac{a_{ij} - \sum\limits_{k=1}^{i-1} l_{ik} u_{kj}}{l_{ii}}. \tag{2.12}$$

   - Calcul de l_{nn} :

$$l_{nn} = a_{nn} - \sum_{k=1}^{n-1} l_{nk} u_{kn} \tag{2.13}$$

2. Descente et remontée triangulaires
   - Descente triangulaire pour résoudre $Ly = b$:

$$y_1 = \frac{b_1}{l_{11}}.$$

   - Pour $i = 2, 3, 4, ..., n$:

$$y_i = \frac{b_i - \sum\limits_{k=1}^{i-1} l_{ik} y_k}{l_{ii}}. \tag{2.14}$$

– Remontée triangulaire pour résoudre $Ux = y(u_{ii} = 1)$:

$$x_n = y_n$$

Pour $i = n-1, n-2, ...2, 1$:

$$x_i = y_i - \sum_{k=i+1}^{n} u_{ik} x_k. \tag{2.15}$$

Remarque 2.4.2. *L'algorithme précédent ne fonctionne que si les pivots l_{ii} sont tous non nuls. Ce n'est pas toujours le cas et il est possible qu'il faille permuter deux lignes pour éviter cette situation, tout comme pour l'élimination de Gauss. Le coefficient l_{ii} est encore appelé pivot. Nous abordons un peu plus loin les techniques de recherche du meilleur pivot.*

Définition 2.4.1. *La notation compacte de la décomposition LU est la matrice de coefficients :*

$$\begin{pmatrix} l_{11} & u_{12} & u_{13} & u_{14} \\ l_{21} & l_{22} & u_{23} & u_{24} \\ l_{31} & l_{32} & l_{33} & u_{34} \\ l_{41} & l_{42} & l_{43} & l_{44} \end{pmatrix} \tag{2.16}$$

dans le cas d'une matrice de dimension 4 sur 4. La matrice initiale A est tout simplement détruite. Les coefficients 1 sur la diagonale de la matrice U ne sont pas indiqués explicitement, mais doivent tout de même être pris en compte. De façon plus rigoureuse, la notation compacte revient à mettre en mémoire la matrice : $L + U - I$ et à détruire la matrice A.

Exemple 2.4.2. *Soit le système :*

$$\begin{pmatrix} 3 & -1 & 2 \\ 1 & 2 & 3 \\ 2 & -2 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 12 \\ 11 \\ 2 \end{pmatrix}$$

que l'on doit décomposer en un produit LU. Pour illustrer la notation compacte, on remplace au fur et à mesure les coefficients a_{ij} par les coefficients l_{ij} ou u_{ij} ; les cases soulignent que l'élément a_{ij} correspondant a été détruit.

1. *Première colonne de L.*
   C'est tout simplement la première colonne de A :

$$\begin{pmatrix} 3 & -1 & 2 \\ 1 & 2 & 3 \\ 2 & -2 & -1 \end{pmatrix}.$$

2. *Première ligne de U.*
   Le pivot de la première ligne est 3. On divise donc la première ligne de A par 3 :

$$\begin{pmatrix} 3 & -1/3 & 2/3 \\ 1 & 2 & 3 \\ 2 & -2 & -1 \end{pmatrix}$$

3. *Deuxième colonne de L De la relation 2.11, on tire :*

$$l_{22} = a_{22} - l_{21}u_{12} = 2 - (1)(-\frac{1}{3}) = \frac{7}{3}$$
$$l_{32} = a_{32} - l_{31}u_{12} = -2 - (2)(-\frac{1}{3}) = -\frac{4}{3}.$$

   On a maintenant :

$$\begin{pmatrix} 3 & -1/3 & 2/3 \\ 1 & 7/3 & 3 \\ 2 & -4/3 & -1 \end{pmatrix}.$$

4. *Deuxième ligne de U.*
   De la relation 2.12, on tire :

$$u_{23} = \frac{a_{23} - l_{21}u_{13}}{l_{22}} = \frac{3 - (1)(\frac{2}{3})}{\frac{7}{3}} = 1.$$

   La matrice compacte devient :

$$\begin{pmatrix} 3 & -1/3 & 2/3 \\ 1 & 7/3 & 1 \\ 2 & -4/3 & -1 \end{pmatrix}.$$

5. *Calcul de l_{33}*
   D'après la relation 2.13, on a :

$$l_{33} = a_{33} - l_{31}u_{13} - l_{32}u_{23} = -1 - (2)(\frac{2}{3}) - (-\frac{4}{3})(1) = -1.$$

   La matrice compacte est donc :

$$\begin{pmatrix} 3 & -1/3 & 2/3 \\ 1 & 7/3 & 1 \\ 2 & -4/3 & -1 \end{pmatrix}.$$

   La matrice de départ A (maintenant détruite) vérifie nécessairement :

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

6. *Résolution de $Ly = b$.*
   La descente triangulaire donne :

$$\begin{array}{rclcrcl} y_1 & = & \dfrac{b_1}{l_{11}} & = & 123 & = & 4 \\ y_2 & = & \dfrac{b_2 - l_{21}y_1}{l_{22}} & = & \dfrac{11 - (1)(4)}{\frac{7}{3}} & = & 3 \\ y_3 & = & \dfrac{b_3 - l_{31}y_1 - l_{32}y_2}{l_{33}} & = & \dfrac{2 - (2)(4) - (-\frac{4}{3})(3)}{(-1)} & = & 2. \end{array}$$

7. *Résolution de $Ux = y$*

$$\begin{array}{rclcrcl} x_3 & = & y_3 & = & 2 & & \\ x_2 & = & y_2 - u_{23}x_3 & = & 3 - (1)(2) & = & 1 \\ x_1 & = & y_1 - u_{12}x_2 - u_{13}x_3 & = & 4 - (-1/3)(1) - (2/3)(2) & = & 3. \end{array}$$

   La solution recherchée est donc

$$x = [3 \;\; 1 \;\; 2]^t.$$

2.4.3 Décomposition LU et permutation de lignes

Comme nous l'avons déjà remarqué, l'algorithme de décomposition LU exige que les pivots l_{ii} soient non nuls. Dans le cas contraire, il faut essayer de permuter deux lignes. Contrairement à la méthode d'élimination de Gauss, la décomposition LU n'utilise le terme de droite b qu'à la toute fin, au moment de la descente triangulaire $Ly = b$. Si l'on permute des lignes, on doit en garder la trace de façon à effectuer les mêmes permutations sur b. À cette fin, on introduit un vecteur O dit de permutation qui contient tout simplement la numérotation des équations.

Remarque 2.4.3. *Dans une décomposition LU, la permutation de lignes s'effectue toujours après le calcul de chaque colonne de L. On place en position de pivot le plus grand terme en valeur absolue de cette colonne (sous le pivot actuel), pour des raisons de précision que nous verrons plus loin.*

Illustrons cela par un exemple.

Exemple 2.4.3. *Soit :*

$$\begin{pmatrix} 0 & 2 & 1 \\ 1 & 0 & 0 \\ 3 & 0 & 1 \end{pmatrix} \overrightarrow{O} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}.$$

Au départ, le vecteur $\overrightarrow{O}$ indique que la numérotation des équations n'a pas encore été modifiée.

1. *Première colonne de L.*
   Puisqu'il s'agit de la première colonne de A, on a :

$$\begin{pmatrix} 0 & 2 & 1 \\ 1 & 0 & 0 \\ 3 & 0 & 1 \end{pmatrix} \overrightarrow{O} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

   Le vecteur de permutation n'a pas été modifié, mais on a un pivot nul. On effectue alors l'opération $(l_1 \longleftrightarrow l_3)$. On aurait tout aussi bien pu permuter la ligne 1 et la ligne 2, mais on choisit immédiatement le plus grand pivot possible (en valeur absolue). Le vecteur de permutation est alors modifié :

$$\begin{pmatrix} 3 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 2 & 1 \end{pmatrix} \overrightarrow{O} = \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix}$$

2. *Première ligne de U Il suffit de diviser cette ligne par le nouveau pivot 3 :*

$$\begin{pmatrix} 3 & 0 & 1/3 \\ 1 & 0 & 0 \\ 0 & 2 & 1 \end{pmatrix} \overrightarrow{O} = \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix}.$$

3. *Deuxième colonne de L De la relation 2.11, on tire :*

$$\begin{array}{rcccccl} l_{22} & = & a_{22} - l_{21}u_{12} & = & 0 - (1)(0) & = & 0 \\ l_{32} & = & a_{32} - l_{31}u_{12} & = & 2 - (0)(0) & = & 2. \end{array}$$

   On a maintenant :

$$\begin{pmatrix} 3 & 0 & 1/3 \\ 1 & 0 & 0 \\ 0 & 2 & 1 \end{pmatrix} \overrightarrow{O} = \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix}$$

   et encore un pivot nul, qui oblige à intervertir les lignes 2 et 3 et à modifier $\overrightarrow{O}$ en conséquence $(l_2 \longleftrightarrow l_3)$:

$$\begin{pmatrix} 3 & 0 & 1/3 \\ 0 & 2 & 1 \\ 1 & 0 & 0 \end{pmatrix} \overrightarrow{O} = \begin{pmatrix} 3 \\ 1 \\ 2 \end{pmatrix}.$$

4. *Calcul de u_{23}*
   La relation 2.12 mène à :

$$u_{23} = \frac{a_{23} - l_{21}u_{13}}{l_{22}} = \frac{1 - (0)(\frac{1}{3})}{2} = \frac{1}{2}.$$

et la matrice compacte devient :

$$\begin{pmatrix} 3 & 0 & 1/3 \\ 0 & 2 & 1/2 \\ 1 & 0 & 0 \end{pmatrix} \overrightarrow{O} = \begin{pmatrix} 3 \\ 1 \\ 2 \end{pmatrix}.$$

5. *Calcul de l_{33} On calcule enfin :*

$$l_{33} \;=\; a_{33} - l_{31}u_{13} - l_{32}u_{23} \;=\; 0 - (1)(\frac{1}{3}) - (0)(\frac{1}{2}) \;=\; -\frac{1}{3}.$$

La décomposition LU de la matrice A est donc :

$$\begin{pmatrix} 3 & 0 & 1/3 \\ 0 & 2 & 1/2 \\ 1 & 0 & -1/3 \end{pmatrix} \overrightarrow{O} = \begin{pmatrix} 3 \\ 1 \\ 2 \end{pmatrix}$$

Il faut toutefois remarquer que le produit LU donne :

$$\begin{pmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 1 & 0 & -1/3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1/3 \\ 0 & 1 & 1/2 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 3 & 0 & 1 \\ 0 & 2 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

c'est-à-dire la matrice A permutée suivant le vecteur $\overrightarrow{O}$. On veut maintenant résoudre : $Ax = \begin{pmatrix} 5 \\ -1 \\ -2 \end{pmatrix}$ Compte tenu du vecteur $\overrightarrow{O}$, on résout d'abord : $Ly = \begin{pmatrix} -2 \\ 5 \\ -1 \end{pmatrix}$ À noter l'ordre des valeurs dans le membre de droite. La descente triangulaire (laissée en exercice) donne

$$y = [-\frac{2}{3} \quad \frac{5}{2} \quad 1]^t.$$

Il suffit maintenant d'effectuer la remontée triangulaire :

$$Ux = \begin{pmatrix} -\frac{2}{3} & \frac{5}{2} & 1 \end{pmatrix}^t$$

qui nous donne la solution finale

$$x = [-1 \quad 2 \quad 1]^t.$$

.

Remarque 2.4.4. *Le déterminant de la matrice A de l'exemple précédent est donné par :*

$$detA = (-1)(-1)[(3)(2)(-\frac{1}{3})] = -2$$

Comme on a permuté deux lignes deux fois, le déterminant a changé de signe deux fois.

Cela nous amène au théorème suivant.

Théorème 2.4.1. *On peut calculer le déterminant d'une matrice A à l'aide de la méthode de décomposition LU de Crout de la façon suivante :*

$$detA = (-1)^N \prod_{i=1}^{n} l_{ii} \tag{2.17}$$

où N est le nombre de fois où on a interverti deux lignes.

Théorème 2.4.2. *Une décomposition LU pour la résolution d'un système linéaire de dimension n sur n requiert exactement* $\frac{n^3-n}{3}$ *multiplications/divisions et* $\frac{2n3-3n2+n}{6}$ *additions/soustractions à l'étape de décomposition en un produit LU . De plus, les remontée et descente triangulaires nécessitent* n^2 *multiplications/divisions et* (n^2-n) *additions/soustractions, pour un total de :*

$$n^3+3n^2-n^3 \quad \textit{multiplications/divisions}$$

et :

$$\frac{2n^3+3n^2-5n}{6} \quad \textit{additions/soustractions.}$$

2.4.4 Factorisation de Choleski

Dans le cas où la matrice est symétrique, on peut diminuer l'espace mémoire nécessaire à la résolution d'un système linéaire. Dans un premier temps, la matrice étant symétrique, on peut se limiter à ne mettre en mémoire que la moitié inférieure de la matrice plus sa diagonale principale. On peut ensuite recourir à la factorisation de Choleski qui consiste à décomposer A sous la forme LL^t où L est encore ici une matrice triangulaire inférieure. La matrice triangulaire supérieure U de la décomposition LU est ainsi remplacée par la matrice transposée de L, réduisant ainsi l'espace mémoire nécessaire. Si on souhaite résoudre un système linéaire, on procède encore ici en deux étapes. On résout d'abord le système triangulaire inférieur $Ly = b$ et ensuite le système triangulaire supérieur $L^t x = y$.

Factorisation de Choleski

1. Premier pivot :
$$l_{11} = \sqrt{a_{11}};$$
2. Première colonne : pour i allant de 2 à n :
$$l_{i1} = a_{i1}/l_{11};$$
3. Pour k allant de 2 à n :
   - Terme diagonal (pivot) :
   $$l_{kk} = \sqrt{a_{kk} - \sum_{j=1}^{k-1} l_{kj}^2}.$$
   - Reste de la colonne :
   $$l_{ik} = \frac{a_{ik} - \sum_{j=1}^{k-1} l_{ij} l_{kj}}{l_{kk}},$$
   pour i allant de $k+1$ à n

Remarque 2.4.5. *On note que la moitié supérieure de la matrice A n'est jamais utilisée et par conséquent, elle n'est pas mise en mémoire inutilement. On sauve ainsi près de la moitié l'espace mémoire par rapport à une matrice non symétrique. Tout comme nous l'avons fait pour la décomposition LU, on peut remplacer, au fur et à mesure que les calculs progressent, les éléments utilisés de la matrice A par l'élément correspondant de L.*

Remarque 2.4.6. *La factorisation de Choleski n'est pas unique. On a par exemple* $A = LL^t = (-L)(-L)^t$ *qui sont deux factorisations différentes. On peut s'assurer de l'unicité en imposant* $l_{ii} > 0$*, ce qui revient au choix naturel de prendre la valeur positive de la racine carrée lors du calcul de* l_{ii}*. Notons enfin que le déterminant de A est donné par :*

$$\det(A) = \det(L)\det(L^t) = (\det(L))^2 = \prod_{i=1}^{n} l_{ii}^2.$$

Exemple 2.4.4. *Considérons la matrice symétrique :*

$$A = \begin{pmatrix} 4 & * & * \\ 6 & 10 & * \\ 2 & 5 & 14 \end{pmatrix}.$$

On a remplacé la partie supérieure de la matrice par des $*$ simplement pour indiquer que cette partie de la matrice n'est pas mise en mémoire et ne servira aucunement dans les calculs. En suivant l'algorithme précédent :

$$l_{11} = \sqrt{a_{11}} = \sqrt{4} = 2,$$

$$l_{21} = \frac{a_{21}}{l_{11}} = 3,$$

$$l_{31} = \frac{a_{31}}{l_{11}} = 1.$$

Sous forme compacte, on a :

$$\begin{pmatrix} 2 & * & * \\ 3 & 10 & * \\ 1 & 5 & 14 \end{pmatrix}.$$

Pour la deuxième colonne :

$$l_{22} = \sqrt{a_{22} - l_{21}^2} = \sqrt{10 - 3^2} = 1,$$

$$l_{32} = \frac{a_{32} - l_{31}l_{21}}{l_{22}} = \frac{5 - (1)(3)}{1} = 2$$

ce qui donne la forme compacte :

$$\begin{pmatrix} 2 & * & * \\ 3 & 1 & * \\ 1 & 2 & 14 \end{pmatrix}.$$

Enfin :

$$l_{33} = \sqrt{a_{33} - l_{31}^2 - l_{32}^2} = \sqrt{14 - (1)^2 - (2)^2} = 3$$

et on a la factorisation sous forme compacte :

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

qui revient à écrire :

$$A = \begin{pmatrix} 4 & 6 & 2 \\ 6 & 10 & 5 \\ 2 & 5 & 14 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 \\ 3 & 1 & 0 \\ 1 & 2 & 3 \end{pmatrix} \begin{pmatrix} 2 & 3 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 3 \end{pmatrix}.$$

On ne peut malheureusement pas appliquer la factorisation de Choleski à toutes les matrices symétriques et pour s'en convaincre, il suffit de considérer la matrice : $A = \begin{pmatrix} -1 & 2 \\ 2 & 6 \end{pmatrix}$. L'algorithme s'arrête dès la toute première étape puisque a 11 est négatif et que l'on ne peut en extraire la racine carrée. Les matrices symétriques pour lesquelles la factorisation de Choleski peut être menée à terme sont dites définies positives.

Définition 2.4.2. *Une matrice symétrique est dite définie positive si :*

$$Ax \,.\, x > 0, \quad \forall x \neq 0.$$

Une matrice est donc définie positive si le produit scalaire de tout vecteur x avec Ax est strictement positif. On peut caractériser les matrices définies positives de bien des façons différentes et équivalentes.

Théorème 2.4.3. *Les énoncés suivants sont équivalents :*

1. *A est une matrice symétrique et définie positive ;*
2. *Toutes les valeurs propres de A sont réelles et strictement positives ;*
3. *Le déterminant des sous-matrices principales de A est strictement positif ;*
4. *Il existe une factorisation de Choleski $A = LL^t$.*

2.4.5 Les systèmes tridiagonaux

Définition 2.4.3. *Une matrice est dite tridiagonale si ses seuls termes non nuls sont situés sur la diagonale principale et les deux diagonales adjacentes. Tous les autres termes sont alors nuls et elle prend la forme :*

$$\begin{pmatrix} a_{11} & a_{12} & 0 & ... & ... & 0 \\ a_{21} & a_{22} & a_{23} & 0 & & ... \\ 0 & a_{32} & a_{33} & ... & ... & ... \\ ... & ... & ... & ... & ... & 0 \\ ... & & ... & ... & a_{(n-1)(n-1)} & a_{(n-1)n} \\ 0 & ... & ... & 0 & a_{n(n-1)} & a_{nn} \end{pmatrix}$$

ou encore

$$\begin{pmatrix} D_1 & S_1 & 0 & ... & 0 \\ I_1 & D_2 & S_2 & ... & ... \\ 0 & I_2 & D_3 & ... & 0 \\ ... & ... & ... & ... & S_{n-1} \\ 0 & ... & 0 & I_{n-1} & D_n \end{pmatrix}.$$

Les systèmes tridiagonaux sont particulièrement faciles à résoudre puisque la décomposition LUe réduit dans ce cas à peu d'opérations. On se réfère donc à la factorisation de Crout qui prendra la forme suivante :

$$\begin{pmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & a_{23} \\ 0 & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ 0 & l_{32} & l_{33} \end{pmatrix} \begin{pmatrix} 1 & u_{12} & 0 \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{pmatrix}.$$

Une colonne de L étant complétée, la ligne de U se réduit à :

$$u_{i-1i} = \frac{a_{i-1i} - \sum_{k=1}^{i-2} l_{i-1k}u_{ki}}{l_{i-1i-1}} = \frac{a_{i-1i}}{l_{i-1i-1}} \left(S_{i-1} \longleftarrow \frac{S_{i-1}}{D_{i-1}} \right).$$

On passe ensuite au pivot :

$$l_{ii} = a_{ii} - \sum_{k=1}^{i-1} l_{ik}u_{ki} = a_{ii} - l_{ii-1}u_{i-1i} \, (D_i \longleftarrow D_i - I_{i-1}S_{i-1})$$

et à la colonne de L réduite à un seul terme :

$$l_{i+1i} = a_{i+1i} - \sum_{k=1}^{i-1} l_{jk}u_{ki} = a_{i+1i}.$$

La diagonale inférieure de A n'est donc pas modifiée par la factorisation. La résolution de $Ly = b$ commence par $y_1 = b_1/l_{11} \quad (y_1 = b_1/D_1)$ et par la suite :

$$y_i = \frac{b_i - \sum_{k=1}^{i-1} l_{ik} y_k}{l_{ii}} = \frac{b_i - l_{ii-1} y_{i-1}}{l_{ii}} \quad \left(y_i \longleftarrow \frac{b_i - I_{i-1} y_{i-1}}{D_i} \right).$$

On constate que cette descente triangulaire peut être effectuée immédiatement après le calcul des colonnes de L. Enfin, la remontée débute par $x_n = y_n$ et se poursuit par :

$$x_i = y_i - \sum_{k=i+1}^{n} u_{ik} x_k = y_i - u_{ii+1} x_{i+1} \quad \left(x_i \longleftarrow \frac{y_i - S_i x_{i+1}}{D_i} \right).$$

2.5 Calcul de la matrice inverse A^{-1}

Exemple 2.5.1. *On doit calculer l'inverse de la matrice :*

$$\begin{pmatrix} 0 & 2 & 1 \\ 1 & 0 & 0 \\ 3 & 0 & 1 \end{pmatrix}$$

dont nous avons déjà obtenu la décomposition LU :

$$\begin{pmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 1 & 0 & -1/3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1/3 \\ 0 & 1 & 1/2 \\ 0 & 0 & 1 \end{pmatrix} \; ; \; \overrightarrow{O} = \begin{pmatrix} 3 \\ 1 \\ 2 \end{pmatrix}.$$

On a recours encore une fois au vecteur de permutation $\overrightarrow{O}$. Pour obtenir la matrice inverse de A, on doit résoudre les trois systèmes linéaires suivants : $Ac_1 = e_1 Ac_2 = e_2 Ac_3 = e_3$ dont le résultat nous donne les trois colonnes de la matrice A^{-1}. Le premier système est résolu d'abord par la descente triangulaire :

$$\begin{pmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 1 & 0 & -1/3 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}.$$

Il faut prendre garde ici au membre de droite. Il s'agit bien du vecteur $e_1 = [1\ 0\ 0]^t$, mais ordonné suivant le vecteur $\overrightarrow{O} = [3\ 1\ 2]^t$ pour tenir compte des lignes qui ont été permutées lors de la décomposition LU. La réso-lution conduit à $y = [0\ \frac{1}{2}\ 0]^t$. Il reste à effectuer la remontée triangulaire :

$$\begin{pmatrix} 1 & 0 & 1/3 \\ 0 & 1 & 1/2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{1}{2} \\ 0 \end{pmatrix}$$

dont le résultat $[0\ \frac{1}{2}\ 0]^t$ représente la première colonne de A^{-1}. Le deuxième système exige dans un premier temps la résolution de :

$$\begin{pmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 1 & 0 & -1/3 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

(à surveiller l'ordre des composantes du vecteur e_2 à droite), dont la solution est $y = [0\ 0\ -3]^t$. Par la suite :

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

qui donne la deuxième colonne de A^{-1}, *soit* $c_2 = [1 \quad \frac{3}{2} \quad 3]^t$. *Enfin, un raisonnement similaire détermine la troisième colonne* $c_3 = [0 \quad -\frac{1}{2} \quad 1]^t$. *La matrice inverse est donc :*

$$A^{-1} = \begin{pmatrix} 0 & 1 & 0 \\ \frac{1}{2} & \frac{3}{2} & -\frac{1}{2} \\ 0 & -3 & 1 \end{pmatrix}.$$

Exemple 2.5.2. *Cet exemple illustre comment effectuer une permutation de façon systématique. Seules les grandes étapes de la décomposition sont indiquées, les calculs étant laissés en exercice. Les coefficients de la matrice sont détruits au fur et à mesure que les calculs progressent et sont remplacés par* l_{ij} *ou* u_{ij}. *Considérons donc la matrice :*

$$\begin{pmatrix} 1 & 6 & 9 \\ 2 & 1 & 2 \\ 3 & 6 & 9 \end{pmatrix} \; ; \; \overrightarrow{O} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}.$$

La première colonne de L *étant la première colonne de* A, *on a :*

$$\begin{pmatrix} 1 & 6 & 9 \\ 2 & 1 & 2 \\ 3 & 6 & 9 \end{pmatrix} \; ; \; \overrightarrow{O} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}.$$

On peut alors permuter la ligne 3 *et la ligne* 1 *de manière à placer en position de pivot le plus grand terme de la première colonne de* L. *On a maintenant :*

$$\begin{pmatrix} 3 & 6 & 9 \\ 2 & 1 & 2 \\ 1 & 6 & 9 \end{pmatrix} \; ; \; \overrightarrow{O} = \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix}.$$

On calcule la première ligne de U *:*

$$\begin{pmatrix} 3 & 2 & 3 \\ 2 & 1 & 2 \\ 1 & 6 & 9 \end{pmatrix} \; ; \; \overrightarrow{O} = \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix}.$$

La deuxième colonne de L *devient alors :*

$$\begin{pmatrix} 3 & 2 & 3 \\ 2 & -3 & 2 \\ 1 & 4 & 9 \end{pmatrix} \; ; \; \overrightarrow{O} = \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix}.$$

On voit qu'il faut maintenant permuter les deux dernières lignes pour amener en position de pivot le plus grand terme de la colonne, qui est 4.

$$\begin{pmatrix} 3 & 2 & 3 \\ 1 & 4 & 9 \\ 2 & -3 & 2 \end{pmatrix} \; ; \; \overrightarrow{O} = \begin{pmatrix} 3 \\ 1 \\ 2 \end{pmatrix}.$$

En continuant ainsi, on trouve la décomposition LU *sous forme compacte :*

$$\begin{pmatrix} 3 & 2 & 3 \\ 1 & 4 & 3/2 \\ 2 & -3 & 1/2 \end{pmatrix} \; ; \; \overrightarrow{O} = \begin{pmatrix} 3 \\ 1 \\ 2 \end{pmatrix}.$$

Chapitre 3

Méthodes itératives

3.1 Définitions de quelques normes

Définition 3.1.1. *Une norme vectorielle est une application de $\mathbb{R}^n$ dans $\mathbb{R}$ ($\mathbb{R}$ désigne l'ensemble des réels) qui associe à un vecteur x un scalaire noté $\|\overrightarrow{x}\|$ et qui vérifie les trois propriétés suivantes :*

1. La norme d'un vecteur est toujours strictement positive, sauf si le vecteur a toutes ses composantes nulles :

$$\|\overrightarrow{x}\| > 0, \quad \textit{sauf si} \quad \overrightarrow{x} = 0. \tag{3.1}$$

2. Si a est un scalaire, alors :

$$\|a\overrightarrow{x}\| = |a|\|\overrightarrow{x}\| \tag{3.2}$$

où $|a|$ est la valeur absolue de a.

3. L'inégalité triangulaire est toujours vérifiée entre deux vecteurs ?x et ?y quelconques :

$$\|\overrightarrow{x} + \overrightarrow{y}\| \leq \|\overrightarrow{x}\| + \|y\|. \tag{3.3}$$

Toute application vérifiant ces trois propriétés est une norme vectorielle. La plus connue est sans doute la norme euclidienne.

Définition 3.1.2. *La norme euclidienne d'un vecteur $\overrightarrow{x}$ est notée $\|\overrightarrow{x}\|_2$ et est définie par :*

$$\|\overrightarrow{x}\|_2 = \sqrt{x_1^2 + x_2^2 + ... + x_n^2} \tag{3.4}$$

Définition 3.1.3. *La norme L_1 est définie par $\|\overrightarrow{x}\|_1 = \sum_{i=1}^{n} |x_i|$ tandis que la norme L_∞ est définie par $\|\overrightarrow{x}\|_\infty = \max_{1 \leq i \leq n} |x_i|$.*

Exemple 3.1.1. *Pour le vecteur $\overrightarrow{x} = [1 \quad -3 \quad -8]^t$, on a :*

$\|\overrightarrow{x}\|_1 = 1 + 3 + 8 = 12$

$\|\overrightarrow{x}\|_\infty = max(1, 3, 8) = 8$

$\|\overrightarrow{x}\|_2 = \sqrt{1 + 9 + 64} = \sqrt{74}$.

Définition 3.1.4. *Une norme matricielle est une application qui associe à une matrice A un scalaire noté $\|A\|$ vérifiant les quatre propriétés suivantes :*

1. *La norme d'une matrice est toujours strictement positive, sauf si la matrice a toutes ses composantes nulles :*

$$||A|| > 0, \ \textit{sauf si } A = 0. \tag{3.5}$$

2. *Si a est un scalaire, alors :*
$$\|aA\| = |a|\|A\|. \tag{3.6}$$

3. *L'inégalité triangulaire est toujours vérifiée entre deux matrices A et B quelconques, c'est-à-dire :*
$$\|A + B\| \leq \|A\| + \|B\| \tag{3.7}$$

4. *Une quatrième propriété est nécessaire pour les matrices :*
$$\|AB\| \leq \|A\|\|B\|. \tag{3.8}$$

Théorème 3.1.1. *Les normes matricielles induites par les normes vectorielles L_1 et L_∞ sont respectivement données par :*

$$\|A\|_1 = \sup_{\|\overrightarrow{x}\|_1=1} \|A\overrightarrow{x}\|_1 = \max_{1\leq j\leq n} \left(\sum_{i=1}^{n} |a_{ij}| \right),$$

$$\|A\|_\infty = \sup_{\|x\|_\infty=1} \|A\overrightarrow{x}\|_\infty = \max_{1\leq i\leq n} \left(\sum_{j=1}^{n} |a_{ij}| \right).$$

La norme $\|A\|_1$ consiste à sommer (en valeur absolue) chacune des colonnes de A et à choisir la plus grande somme. La norme $\|A\|_\infty$ fait un travail similaire sur les lignes.

Remarque 3.1.1. *On définit la norme de Frobenius par :*

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

On peut montrer que la norme de Frobenius n'est induite par aucune norme vectorielle. Par contre, on peut aussi montrer que la norme matricielle induite par la norme euclidienne est :

$$\|A\|_2 = \sup_{\|\overrightarrow{x}\|_2=1} \|A\overrightarrow{x}\|_2 = \sqrt{\rho(A^tA)}$$

où ρ désigne le rayon spectral, c'est-à-dire la plus grande valeur propre.

Exemple 3.1.2. *Soit la matrice :*
$$\begin{pmatrix} 1 & -2 & 5 \\ -3 & 1 & -5 \\ 1 & -9 & 0 \end{pmatrix}.$$

Les différentes normes prennent alors les valeurs suivantes :
$\|A\|_1 = max(5, 12, 10) = 12$
$\|A\|_\infty = max(8, 9, 10) = 10$
$\|A\|_F = \sqrt{1 + 4 + 25 + 9 + 1 + 25 + 1 + 81} = \sqrt{147}.$

Définition 3.1.5. *Une norme vectorielle et une norme matricielle sont dites compatibles si la condition :*
$$\|A\overrightarrow{x}\| \leq \|A\|\|\overrightarrow{x}\| \tag{3.9}$$
est valide quels que soient la matrice A et le vecteur $\overrightarrow{x}$.

Exemple 3.1.3. *Considérons de nouveau le vecteur* $\overrightarrow{x} = [1 \quad -3 \quad -8]^t$ *et la matrice :*

$$\begin{pmatrix} 1 & -2 & 5 \\ -3 & 1 & -5 \\ 1 & -9 & 0 \end{pmatrix}.$$

Le produit $A\overrightarrow{x}$ *donne le vecteur* $[-33 \quad 34 \quad 28]^t$ *et donc :*

$\|A\overrightarrow{x}\|_1 = 95, \quad \|A\overrightarrow{x}\|_\infty = 34 \quad$ *et* $\quad \|A\overrightarrow{x}\|_2 = \sqrt{3029}.$

L'inégalité 3.9 devient respectivement :

$$\begin{array}{rcll} 95 & \leq & (12)(12) & \text{en norme } L_1 \\ 34 & \leq & (10)(8) & \text{en norme } L_\infty \\ \sqrt{3029} & \leq & (\sqrt{147})(\sqrt{74}) = \sqrt{10878} & \text{en norme euclidienne.} \end{array}$$

3.2 Principe général des méthodes itératives

Le principe de base de telles méthodes est d'engendrer une suite de vecteurs x_k (les itérés) convergente vers la solution x du système linéaire $Ax = b$.

Pour cela on décompose la matrice A sous la forme $A = M - N$ où la matrice M est inversible. Ainsi, le système $Ax = b$ donne $Mx = Nx + b$ et donc $x = M^{-1}Nx + M^{-1}b$. En posant $B = M^{-1}N$ et $c = M^{-1}b$, il est claire que le système $Ax = b$ est équivalent au problème de point fixe suivant : "chercher $x \in \mathbb{R}^n$ tel que $x = Bx + c$.

En somme, la plupart des méthodes itératives sont de la forme suivante : Partant d'un vecteur arbitraire x_0, on engendre une suite $(x_k)_k$ définie par

$$x_{k+1} = Bx_k + c \tag{3.10}$$

avec B une matrice $\mathcal{M}_{n\times n}(K)$, $c \in K^n$; avec $K = \mathbb{R}$ ou $\mathbb{C}$.

Définition 3.2.1. *Une méthode itérative de la forme (3.10) est dite convergente si pour tout* x_0, *on a* $x_k \longrightarrow x$, *quand* $k \longrightarrow +\infty$ *et la limite vérifie* $Ax = b$. *(*$Ax = b$ *est équivalent alors à* $x = Bx + c$*).*

Définition 3.2.2. *L'erreur d'approximation à la kième étape s'écrit* $e_k = x_k - x = Bx_{k-1} + c - Bx - c = B(x_{k-1} - x) = Be_{k-1}$. *Et aussi* $e_k = B^k e_0, \ \forall k \in \mathbb{N}$.

Définition 3.2.3. *Une méthode itérative est convergente si pour tout* x_0, *on a* $\lim\limits_{k\to\infty} e_k = 0$. *Ceci est équivalent à :* $\left(\lim\limits_{k\to\infty} B^k = 0\right) \Longleftrightarrow \left(\forall x \in K^n, \lim\limits_{k\to\infty} B^k x = 0\right) \Longleftrightarrow \left(\lim\limits_{k\to\infty} \|B\|^k = 0\right)$ *pour toute norme matricielle.*

Théorème 3.2.1. *(Convergence des méthodes itératives)*

On a $\lim\limits_{k\to\infty} B^k = 0 \Longleftrightarrow \rho(B) < 1$.

Pour qu'une méthode itérative de la forme (3.10) soit convergente il faut et il suffit que $\rho(B) < 1$. *on rappelle que* $\rho(B)$ *désigne le rayon spectrale de la matrice* B *et* $\rho(B) = \max\limits_i |\lambda_i(B)|$.

Preuve 3.2.1. $(\Longrightarrow)$ *si* $\rho(B) \geq 1$, *il existe* λ, *valeur propre de* B, *telle que* $|\lambda| = 1$. *Soit* $x \neq 0$ *vecteur propre associé à* λ, *alors* $B^k x = \lambda^k x$ *et comme* $|\lambda^k| \longrightarrow 1$ *ou* $+\infty$ *alors* $B^k x$ *ne converge pas vers zéro, ainsi* $\lim\limits_{k\to\infty} B^k \neq 0$.

$(\Longleftarrow)$ *On suppose* $\rho(B) < 1$, *c'est à dire que* $|\lambda_n| < 1, \ n = 1, ..., r$ *avec* λ_n *valeur propre de* B. *Toute matrice est semblable à une matrice de Jordan et on conclut.*

Corollaire 3.2.1. *Si $\|B\| < 1$ alors $\rho(B) < 1$ et la méthode (3.10) converge. La preuve est immédiate car $\rho(B) \leq \|B\|$. En effet, soit $Bx = \lambda x$ alors $|\lambda|\|x\| = \|Bx\| \leq \|B\|\|x\|$, soit encore $|\lambda| \leq \|B\|$ pour tout $\lambda \in Sp(B)$.*

Définition 3.2.4. *On appelle taux asymptotique de convergence d'une méthode itérative le nombre $R_\infty = -\log(\rho(B)$. Ce nombre est positif car $\rho(B) < 1$. Ce taux de convergence permet de mesurer le nombre d'itération nécessaire pour réduire l'erreur d'un certain facteur.*

Proposition 3.2.1. *Etant donné $0 < \eta < 1$, si le nombre d'itération $k \geq -\dfrac{\log(\eta)}{R_\infty(B)}$ alors $\|e_k\| \leq \eta\|e_0\|$.*

Preuve 3.2.2. *On a $e_k = B^k e_0$ et $\|e_k\| \leq \|B^k\|\|e_0\|$, $\forall k$. Pour avoir $\dfrac{\|e_k\|}{\|e_0\|} \leq \eta$, on impose $\|B^k\| \leq \eta$, soit $\|B^k\|^{\frac{1}{k}} \leq \eta^{\frac{1}{k}}$ alors $\log(\|B^k\|^{\frac{1}{k}} \leq \dfrac{1}{k}\log(\eta)$, or $\|B^k\| < 1$ alors $k \geq \dfrac{\log(\eta)}{\log\|B^k\|^{\frac{1}{k}}} = \dfrac{-\log(\eta)}{-\log\|B^k\|^{\frac{1}{k}}}$.*

D'autre part, $\rho^k(B) = \rho(B^k) \leq \|B^k\|$, alors $\rho(B) \leq \|B^k\|^{\frac{1}{k}}$,
ainsi $-\log(\rho(B)) = R_\infty(B) \geq -\log(\|B^k\|^{\frac{1}{k}}$ et donc $\dfrac{1}{R_\infty(B)} \leq \dfrac{1}{-\log(\|B^k\|^{\frac{1}{k}})}$.

Enfin, on choisit alors k le nombre d'itérations :

$$k \geq \frac{-\log(\eta)}{R_\infty(B)}$$

pour réduire l'erreur initiale de η.

3.3 Méthodes itératives classiques

Soit A une matrice d'ordre n telle que $a_{ii} \neq 0, \forall i = 1, ..., n$. On décompose A sous la forme

$$A = D - E - F$$

avec D la matrice diagonale de diagonale celle de A ; $-E$ la partie inférieure stricte et $-F$ la partie supérieure stricte.

Remarque 3.3.1. *L'hypothèse $a_{ii} \neq 0, \forall i = 1, ..., n$ permet la mise en place des méthodes suivantes.*

3.3.1 Méthode de Jacobi

Résoudre $Ax = b$ est équivalent à $Dx = (E + F)x + b$. La méthode de Jacobi est basée sur la décomposition précédente et elle s'écrit

$$\begin{cases} x_0 \text{ arbitraire} \\ Dx_{k+1} = (E + F)x_k + b. \end{cases} \tag{3.11}$$

Il est facile à chaque itération de calculer x_{k+1} en fonction de x_k car la matrice diagonale D est inversible. Les composantes du vecteur x_{k+1} vérifient

$$a_{ii}(x_{k+1})_i = -\sum_{j=1, j\neq i}^{n} a_{ij}(x_k)_j + b_i,$$

soit encore

$$(x_{k+1})_i = \frac{1}{a_{ii}}\left(\sum_{j=1, j\neq i}^{n} a_{ij}(x_k)_j + b_i\right).$$

Ce qui se met sous la forme matricielle suivante :

$$x_{k+1} = D^{-1}(E+F)x_k + D^{-1}b = Jx_k + c. \tag{3.12}$$

La matrice $J = D^{-1}(E+F)$ est la matrice d'itération de Jacobi. La méthode de Jacobi converge si seulement si : $\rho(J) < 1$. Pour programmer cette méthode, on a besoin de stocker les vecteurs x_k et x_{k+1}.

3.4 Méthode de Gauss-Seidel

Elle est basée sur cette décomposition : $Ax = b \Longleftrightarrow (D-E)x = Fx + b$ la méthode de Gauss-Seidel s'écrit

$$\begin{cases} x_0 \text{ arbitraire} \\ (D-E)x_{k+1} = Fx_k + b. \end{cases} \tag{3.13}$$

$D-E$ est une matrice triangulaire inférieure et pour calculer x_{k+1} en fonction de x_k, il suffit d'appliquer l'algorithme de descente suivant : pour $i = 1, ..., n$

$$a_{ii}(x_{k+1})_i = -\sum_{j=1}^{n} a_{ij}(x_{k+1})_j - \sum_{j=i+1}^{i-1} a_{ij}(x_k)_j + b_i,$$

soit encore

$$(x_{k+1})_i = \frac{1}{a_{ii}}\left(-\sum_{j=1}^{n} a_{ij}(x_{k+1})_j - \sum_{j=i+1}^{i-1} a_{ij}(x_k)_j + b_i\right).$$

Noter que dans la boucle de calcul, les composantes du vecteur $(x_{k+1})_j$ pour $j = 1, ..., i-1$ sont déjà calculés et on peut utiliser un seul vecteur pour programmer cette méthode. Sous forme matricielle x_{k+1} s'écrit

$$x_{k+1} = (D-E)^{-1}Fx_k + (D-E)^{-1}b = Gx_k + c, \tag{3.14}$$

la matrice $G = (D-E)^{-1}F$ est la matrice d'itération de Gauss-Seidel. La méthode de Gauss-Seidel converge ssi $\rho(G) < 1$.

3.4.1 Méthode de relaxation

Elle est basée sur cette décomposition : Soit $\omega \neq 0$, la matrice A s'écrit

$$A = (\frac{1}{\omega}D - E) + (D - \frac{1}{\omega}D - F).$$

Le système $Ax = b$ s'écrit $(\frac{1}{\omega}D - E)x = ((1 - \frac{1}{\omega})D + F)x + b$. La méthode de relaxation s'écrit :

$$\begin{cases} x_0 \text{ arbitraire} \\ (\frac{1}{\omega}D - E)x_{k+1} = ((1 - \frac{1}{\omega})D + F)x_k + b. \end{cases} \tag{3.15}$$

Ou encore

$$\begin{cases} x_0 \text{ arbitraire} \\ (D - \omega E)x_{k+1} = ((\omega - 1)D + \omega F)x_k + \omega b. \end{cases} \tag{3.16}$$

La matrice d'itération s'écrit alors $\mathcal{R}_\omega = (D - \omega E)^{-1}((1-\omega)D + \omega F)$.

Remarque 3.4.1. *Pour $\omega = 1$, on retrouve la méthode de Gauss-Sidel.*

Théorème 3.4.1. *1. Soit A une matrice symétrique définie positive (ou hermitienne définie positive), alors la méthode de relaxation converge si $0 < \omega < 2$.*

2. Soit A une matrice à diagonale strictement dominante ou à diagonale fortement dominante et irréductible) alors la méthode de Jacobi converge et la méthode de relaxation converge pour $0 < \omega \leq 1$.

3.5 Conditionnement d'une matrice

Définition 3.5.1. *Le conditionnement d'une matrice (noté condA) est défini par :*

$$condA = \|A\|\|A^{-1}\|. \tag{3.17}$$

Il s'agit simplement du produit de la norme de A et de la norme de son inverse.

Remarque 3.5.1. *Le conditionnement dépend de la norme matricielle utilisée. On utilise le plus souvent la norme $\|A\|_\infty$. Il ne reste plus qu'à montrer en quoi le conditionnement d'une matrice est si important pour déterminer la sensibilité d'une matrice aux erreurs d'arrondis et à l'arithmétique flottante. Tout d'abord, on montre que le conditionnement est un nombre supérieur ou égal à 1. En effet, si I désigne la matrice identité (ayant des 1 sur la diagonale et des 0 partout ailleurs), on a :*

$$\|A\| = \|AI\| = \|A\|\|I\|$$

en vertu de la relation 3.8. Cela entraîne, après division par $\|A\|$ de chaque côté, que $\|I\| \geq 1$, quelle que soit la norme matricielle utilisée. On en conclut que :

$$1 \leq \|I\| = \|AA^{-1}\| \leq \|A\|\|A^{-1}\|$$

et donc que :

$$1 \leq condA < \infty. \tag{3.18}$$

.

3.5.1 Bornes d'erreurs et conditionnement

Considérons le système linéaire :

$$A\overrightarrow{x} = \overrightarrow{b}$$

et notons $\overrightarrow{x}$, la solution exacte et $\overrightarrow{x}^*$, une solution approximative qu'on obtient en utilisant l'arithmétique flottante. Ces deux vecteurs devraient être près l'un de l'autre, c'est-à-dire que la norme de l'erreur :

$$\|\overrightarrow{e}\| = \|\overrightarrow{x} - \overrightarrow{x}^*\|$$

devrait être petite. Ce n'est pas toujours le cas. Définissons le résidu par :

$$\overrightarrow{r} = \overrightarrow{b} - A\overrightarrow{x}^*. \tag{3.19}$$

On a alors : $\overrightarrow{r} = \overrightarrow{b} - A\overrightarrow{x}^* = A\overrightarrow{x} - A\overrightarrow{x}^* = A(\overrightarrow{x} - \overrightarrow{x}^*) = A\overrightarrow{e}$ ce qui signifie que $\overrightarrow{e} = A^{-1}\overrightarrow{r}$. Si l'on utilise des normes vectorielles et matricielles compatibles, on a en vertu de la relation 3.9 :

$$\|\overrightarrow{e}\| = \|A^{-1}\|\|\overrightarrow{r}\| \tag{3.20}$$

De façon analogue, puisque $A\overrightarrow{e} = \overrightarrow{r}$:

$$\|\overrightarrow{r}\| = \|A\|\|\overrightarrow{e}\|$$

qui peut s'écrire :

$$\|\overrightarrow{r}\|\|A\| = \|\overrightarrow{e}\|. \tag{3.21}$$

En regroupant les relations 3.20 et 3.21, on obtient :

$$\frac{\|\overrightarrow{r}\|}{\|A\|} \leq \|\overrightarrow{e}\| \leq \|A^{-1}\|\|\overrightarrow{r}\|. \tag{3.22}$$

Par ailleurs, en refaisant le même raisonnement avec les égalités $A\overrightarrow{x} = \overrightarrow{b}$ et $\overrightarrow{x} = A^{-1}\overrightarrow{b}$, on trouve :

$$\|\overrightarrow{b}\|\|A\| = \|\overrightarrow{x}\| = \|A^{-1}\|\|\overrightarrow{b}\|.$$

Après avoir inversé ces inégalités, on trouve :

$$\frac{1}{\|A^{-1}\|\|\overrightarrow{b}\|} \leq \frac{1}{\|\overrightarrow{x}\|} \leq \frac{\|A\|}{\|\overrightarrow{b}\|} \tag{3.23}$$

En multipliant les inégalités 3.22 et 3.23, on obtient le résultat fondamental suivant.

Théorème 3.5.1.

$$\frac{1}{condA}\frac{\|\overrightarrow{r}\|}{\|\overrightarrow{b}\|} \leq \frac{\|\overrightarrow{e}\|}{\|\overrightarrow{x}\|} \leq condA\frac{\|\overrightarrow{r}\|}{\|\overrightarrow{b}\|}. \tag{3.24}$$

Remarque 3.5.2. *Plusieurs remarques s'imposent pour bien comprendre l'inégalité précédente.*

1. *Le terme du milieu représente l'erreur relative entre la solution exacte $\overrightarrow{x}$ et la solution approximative $\overrightarrow{x}^*$.*
2. *Si le conditionnement de la matrice A est près de 1, l'erreur relative est coincée entre deux valeurs très près l'une de l'autre. Si la norme du résidu est petite, l'erreur relative est également petite et la précision de la solution approximative a toutes les chances d'être satisfaisante.*
3. *Par contre, si le conditionnement de la matrice A est grand, la valeur de l'erreur relative est quelque part entre 0 et un nombre possiblement très grand. Il est donc à craindre que l'erreur relative soit alors grande, donc que la solution approximative soit de faible précision et même, dans certains cas, complètement fausse.*
4. *Même si la norme du résidu est petite, il est possible que l'erreur relative liée à la solution approximative soit quand même très grande.*
5. *Plus le conditionnement de la matrice A est grand, plus on doit être attentif à l'algorithme de résolution utilisé.*
6. *Il importe de rappeler que, même si une matrice est bien conditionnée, un mauvais algorithme de résolution peut conduire à des résultats erronés.*
7. *Les deux inégalités de l'équation 3.24 nous permettent d'obtenir une borne inférieure pour le conditionnement. En isolant, on montre en effet que :*

$$condA \geq \frac{\|\overrightarrow{e}\|\|\overrightarrow{b}\|}{\|\overrightarrow{x}\|\|\overrightarrow{r}\|} \quad et \quad condA \geq \frac{\|\overrightarrow{x}\|\|\overrightarrow{r}\|}{\|\overrightarrow{e}\|\|\overrightarrow{b}\|}$$

et donc que :

$$condA \geq max\left(\frac{\|\overrightarrow{e}\|\|\overrightarrow{b}\|}{\|\overrightarrow{x}\|\|\overrightarrow{r}\|}; \frac{\|\overrightarrow{x}\|\|\overrightarrow{r}\|}{\|\overrightarrow{e}\|\|\overrightarrow{b}\|}\right)$$

d'où :

$$condA = max\left(\frac{\|\overrightarrow{x} - \overrightarrow{x}^*\|}{\|\overrightarrow{x}^*\|}\frac{\|\overrightarrow{b}\|}{\|\overrightarrow{b} - A\overrightarrow{x}^*\|}; \frac{\|\overrightarrow{x}\|}{\|\overrightarrow{x} - \overrightarrow{x}^*\|}\frac{\|\overrightarrow{b} - A\overrightarrow{x}^*\|}{\|\overrightarrow{b}\|}\right). \tag{3.25}$$

Cette borne inférieure est valide quel que soit le vecteur $\overrightarrow{x}^*$. Si une ma-trice est mal conditionnée, on peut essayer de déterminer un vecteur $\overrightarrow{x}^*$ pour lequel le terme de droite de l'expression 3.25 sera aussi grand que possible. On peut parfois avoir de cette manière une bonne idée du conditionnement

Théorème 3.5.2.

$$\frac{\|\overrightarrow{x} - \overrightarrow{x}^*\|}{\|\overrightarrow{x}^*\|} \leq condA\frac{\|E\|}{\|A\|}. \tag{3.26}$$

Remarque 3.5.3. *Les remarques suivantes permettent de bien mesurer la portée de l'inégalité 3.26.*

1. *Le terme de gauche est une approximation de l'erreur relative entre la solution exacte et la solution du système perturbé. (On devrait avoir $\|?x\|$ au dénominateur pour représenter vraiment l'erreur relative.)*
2. *Le terme de droite est en quelque sorte l'erreur relative liée aux coefficients de la matrice A multipliée par le conditionnement de A.*
3. *Si condA est petit, une petite perturbation sur la matrice A entraîne une petite perturbation sur la solution $\overrightarrow{x}$.*
4. *Par contre, si condA est grand, une petite perturbation sur la matrice A pourrait résulter en une très grande perturbation sur la solution du système. Il est par conséquent possible que les résultats numériques soient peu précis et même, dans certains cas, complètement faux.*

Remarque 3.5.4. *Très souvent, la perturbation E de la matrice A provient des erreurs dues à la représentation des nombres sur ordinateur. Par définition de la précision machine ?m et de la norme L_∞, on a dans ce cas :*

$$\|E\|_\infty \leq \epsilon_m \|A\|$$

ce qui permet de réécrire la conclusion 3.26 du théorème sous la forme :

$$\|\overrightarrow{x} - \overrightarrow{x}^*\|_\infty \|\overrightarrow{x}^*\|_\infty \leq \epsilon_m cond_\infty A = \epsilon_m \|A\|_\infty \|A - 1\|_\infty. \tag{3.27}$$

On constate que, plus le conditionnement est élevé, plus la précision machine ϵ_m doit être petite. Si la simple précision est insuffisante, on recourt à la double précision.

Exemple 3.5.1. *La matrice :*

$$A = \begin{pmatrix} 1,012 & -2,132 & 3,104 \\ -2,132 & 4,096 & -7,013 \\ 3,104 & -7,013 & 0,014 \end{pmatrix}$$

a comme inverse :

$$A^{-1} = \begin{pmatrix} -13,729 & -6,0755 & 0,62540 \\ -6,0755 & -2,6888 & 0,13399 \\ 0,62540 & 0,13399 & -0,11187 \end{pmatrix}.$$

On a alors $\|A\|_\infty = 13,241$ et $\|A^{-1}\|_\infty = 20,43$. On conclut que le conditionnement de la matrice A est :

$$cond_\infty A = (13,241)(20,43) = 270,51.$$

Exemple 3.5.2. *La matrice :*

$$A = \begin{pmatrix} 3,02 & -1,05 & 2,53 \\ 4,33 & 0,56 & -1,78 \\ -0,83 & -0,54 & 1,47 \end{pmatrix}$$

a comme inverse :

$$A^{-1} = \begin{pmatrix} 5,661 & -7,273 & -18,55 \\ 200,5 & -268,3 & -669,9 \\ 76,85 & -102,6 & -255,9 \end{pmatrix}.$$

Pour cette matrice, $\|A\|_\infty = 6,67$ et $\|A^{-1}\|_\infty = 1138,7$. Le conditionnement de la matrice est donc 7595, ce qui est le fait d'une matrice mal conditionnée.

3.6 Systèmes non linéaires

Le problème consiste à trouver le ou les vecteurs $\overrightarrow{x} = [x_1 \quad x_2 \quad x_3 \quad ... \quad x_n]^t$ vérifiant les n équations non linéaires suivantes :

$$\begin{cases} f_1(x_1, x_2, x_3, ..., x_n) = 0 \\ f_2(x_1, x_2, x_3, ..., x_n) = 0 \\ f_3(x_1, x_2, x_3, ..., x_n) = 0 \\ .... \\ f_n(x_1, x_2, x_3, ..., x_n) = 0 \end{cases} \tag{3.28}$$

où les f_i sont des fonctions de n variables que nous supposons différentiables. Contrairement aux systèmes linéaires, il n'y a pas de condition simple associée aux systèmes non linéaires qui permette d'assurer l'existence et l'unicité de la solution. Le plus souvent, il existe plusieurs solutions possibles et seul le contexte indique laquelle est la bonne. Les méthodes de résolution des systèmes non linéaires sont nombreuses. Notamment, presque toutes les méthodes du chapitre 2 peuvent être générali-sées aux systèmes non linéaires. Pour éviter de surcharger notre exposé, nous ne présentons que la méthode la plus importante et la plus utilisée en pratique, soit la **méthode de Newton**.

L'application de cette méthode à un système de deux équations non li-néaires est suffisante pour illustrer le cas général. Il serait également bon de réviser le développement de la méthode de Newton pour une équation non linéaire (voir le chapitre 2) puisque le raisonnement est le même pour les sys-tèmes. Considérons donc le système :

$$\begin{cases} f_1(x_1, x_2) = 0 \\ f_2(x_1, x_2) = 0 \end{cases}$$

Soit (x_1^0, x_2^0), une approximation initiale de la solution de ce système. Cette approximation initiale est cruciale et doit toujours être choisie avec soin. Le but de ce qui suit est de déterminer une correction $(\delta x_1, \delta x_2)$ à (x_1^0, x_2^0), de telle sorte que :

$$\begin{cases} f_1(x_1^0 + \delta x_1, x_2^0 + \delta x_2) = 0 \\ \\ f_2(x_1^0 + \delta x_1, x_2^0 + \delta x_2) = 0. \end{cases}$$

Pour déterminer $(\delta x_1, \delta x_2)$, il suffit maintenant de faire un développement de Taylor en deux variables pour chacune des deux fonctions (voir Thomas et Finney, réf. [35]) :

$$0 = f_1(x_1^0, x_2^0) + \frac{\partial f_1}{\partial x_1}(x_1^0, x_2^0)\delta x_1 + \frac{\partial f_1}{\partial x_2}(x_1^0, x_2^0)\delta x_2 + ...$$

$$0 = f_2(x_1^0, x_2^0) + \frac{\partial f_2}{\partial x_1}(x_1^0, x_2^0)\delta x_1 + \frac{\partial f_2}{\partial x_2}(x_1^0, x_2^0)\delta x_2 + ...$$

Dans les relations précédentes, les pointillés désignent des termes d'ordre su-périeur ou égal à deux et faisant intervenir les dérivées partielles d'ordre cor-respondant. Pour déterminer $(\delta x_1, \delta x_2)$, il suffit de négliger les termes d'ordre supérieur et d'écrire :

$$\frac{\partial f_1}{\partial x_1}(x_1^0, x_2^0)\delta x_1 + \frac{\partial f_1}{\partial x_2}(x_1^0, x_2^0)\delta x_2 = -f_1(x_1^0, x_2^0)$$

$$\frac{\partial f_2}{\partial x_1}(x_1^0, x_2^0)\delta x_1 + \frac{\partial f_2}{\partial x_2}(x_1^0, x_2^0)\delta x_2 = -f_2(x_1^0, x_2^0)$$

ou encore sous forme matricielle :

$$\begin{pmatrix} \dfrac{\partial f_1}{\partial x_1}(x_1^0, x_2^0) & \dfrac{\partial f_1}{\partial x_2}(x_1^0, x_2^0) \\ \\ \dfrac{\partial f_2}{\partial x_1}(x_1^0, x_2^0) & \dfrac{\partial f_2}{\partial x_2}(x_1^0, x_2^0) \end{pmatrix} \begin{pmatrix} \delta x_1 \\ \\ \delta x_2 \end{pmatrix} = - \begin{pmatrix} f_1(x_1^0, x_2^0) \\ \\ f_2(x_1^0, x_2^0) \end{pmatrix}.$$

Ce système linéaire s'écrit également sous une forme plus compacte :

$$J(x_1^0, x_2^0)\,\overrightarrow{\delta}\,x = -\overrightarrow{R}(x_1^0, x_2^0) \tag{3.29}$$

où $J(x_1^0, x_2^0)$ désigne la matrice des dérivées partielles ou **matrice jacobienne** évaluée au point (x_1^0, x_2^0), où $\overrightarrow{\delta}\,x$ est **le vecteur des corrections** relatives à chaque variable et où $-\overrightarrow{R}(x_1^0, x_2^0)$ est **le vecteur résidu** évalué en (x_1^0, x_2^0). Le **déterminant de la matrice jacobienne est appelé le jacobien.** Le jacobien doit bien entendu être différent de 0 pour que la matrice jacobienne soit inversible. On pose ensuite :

$$x_1^1 = x_1^0 + \delta x_1$$

$$x_2^1 = x_2^0 + \delta x_2$$

qui est la nouvelle approximation de la solution du système non linéaire. On cherchera par la suite à corriger (x_1^1, x_2^1) d'une nouvelle quantité $(\overrightarrow{\delta}\,x)$, et ce, jusqu'à la convergence. De manière plus générale, on pose :

$$J(\overrightarrow{x}^i) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\overrightarrow{x}^i) & \frac{\partial f_1}{\partial x_2}(\overrightarrow{x}^i) & ... & ... & \frac{\partial f_1}{\partial x_n}(\overrightarrow{x}^i) \\ \frac{\partial f_2}{\partial x_1}(\overrightarrow{x}^i) & \frac{\partial f_2}{\partial x_2}(\overrightarrow{x}^i) & ... & ... & \frac{\partial f_2}{\partial x_n}(\overrightarrow{x}^i) \\ ... & ... & ... & ... & \\ \frac{\partial f_n}{\partial x_1}(\overrightarrow{x}^i) & \frac{\partial f_n}{\partial x_2}(\overrightarrow{x}^i) & ... & ... & \frac{\partial f_n}{\partial x_n}(\overrightarrow{x}^i) \end{pmatrix}$$

c'est-à-dire la matrice jacobienne évaluée au point $\overrightarrow{x}^i = (x_1^i, x_2^i, ..., x_n^i)$ De plus, on pose :

$$\overrightarrow{R}(\overrightarrow{x}^i) = \begin{pmatrix} f_1(\overrightarrow{x}^i) \\ f_2(\overrightarrow{x}^i) \\ ... \\ f_n(\overrightarrow{x}^i) \end{pmatrix} \quad \overrightarrow{\delta}\,x = \begin{pmatrix} \delta x_1 \\ \delta x_2 \\ ... \\ \delta x_n \end{pmatrix}$$

pour en arriver à l'algorithme général suivant.

Algorithme : Méthode de Newton appliquée aux systèmes

1. Étant donné ϵ_a, un critère d'arrêt
2. Étant donné N, le nombre maximal d'itérations
3. Étant donné $\overrightarrow{x}^0 = [x_1^0 \;\; x_2^0 \;\; ... \;\; x_n^0]^t$, une approximation initiale de la solution du système
4. Résoudre le système linéaire :
$$J(\overrightarrow{x}^i)\,\overrightarrow{\delta}\,x = -\overrightarrow{R}(\overrightarrow{x}^i) \tag{3.30}$$
et poser : $\overrightarrow{x}^{i+1} = \overrightarrow{x}^i + \overrightarrow{\delta}\,x$
5. Si $\frac{\|\overrightarrow{\delta}\,x\|}{\|\overrightarrow{x}^{i+1}\|} < \epsilon_a$ et $\|\overrightarrow{R}(\overrightarrow{x}^{i+1})\| \leq \epsilon_a$:
   - convergence atteinte
   - écrire la solution $\overrightarrow{x}^{i+1}$
   - arrêt
6. Si le nombre maximal d'itérations N est atteint :
   - convergence non atteinte en N itérations
   - arrêt
7. Retour à l'étape 4

Exemple 3.6.1. *On cherche à trouver l'intersection de la courbe $x_2 = e^{x_1}$ et du cercle de rayon 4 centré à l'origine d'équation $x_1^2 + x_2^2 = 16$. L'intersection de ces courbes est une solution de :*

$$\begin{cases} e^{x_1} - x_2 = 0 \\ x_1^2 + x_2^2 - 16 = 0. \end{cases}$$

La première étape consiste à calculer la matrice jacobienne de dimension 2. Dans ce cas, on a :

$$J(x_1, x_2) = \begin{pmatrix} e^{x_1} & -1 \\ 2x_1 & 2x_2 \end{pmatrix}.$$

La première solution se trouve près du point $(-4\ ;\ 0)$ et la deuxième, près de $(2,8\ ;\ 2,8)$. Prenons le vecteur $\overrightarrow{x}^0 = [2,8\ \ 2,8]^t$ comme approximation initiale de la solution de ce système non linéaire.

1. Itération 1 Le système 3.30 devient :

$$\begin{pmatrix} e^{2,8} & -1 \\ 2(2,8) & 2(2,8) \end{pmatrix} \begin{pmatrix} \delta x_1 \\ \delta x_2 \end{pmatrix} = - \begin{pmatrix} e^{2,8} - 2,8 \\ (2,8)^2 + (2,8)^2 - 16 \end{pmatrix}$$

c'est-à-dire :

$$\begin{pmatrix} 16,445 & -1 \\ 5,6 & 5,6 \end{pmatrix} \begin{pmatrix} \delta x_1 \\ \delta x_2 \end{pmatrix} = - \begin{pmatrix} 13,645 \\ -0,3200 \end{pmatrix}$$

dont la solution est $\overrightarrow{\delta} x = [-0,77890 \quad 0,83604]^t$. La nouvelle approximation de la solution est donc :

$$\begin{array}{rcccccl} x_1^1 & = & x_1^0 + \delta x_1 & = & 2,8 - 0,77890 & = & 2,0211 \\ x_2^1 & = & x_2^0 + \delta x_2 & = & 2,8 + 0,83604 & = & 3,63604. \end{array}$$

2. Itération 2 On effectue une deuxième itération à partir de $[2,0211 \quad 3,63604]^t$. Le système 3.30 devient alors :

$$\begin{pmatrix} e^{2,0211} & -1 \\ 2(2,0211) & 2(3,63604) \end{pmatrix} \begin{pmatrix} \delta x_1 \\ \delta x_2 \end{pmatrix} = - \begin{pmatrix} e^{2,0211} - 3,63604 \\ (2,0211)2 + (3,63604)2 - 16 \end{pmatrix}$$

c'est-à-dire :

$$\begin{pmatrix} 7,5466 & -1 \\ 4,0422 & 7,2721 \end{pmatrix} \begin{pmatrix} \delta x_1 \\ \delta x_2 \end{pmatrix} = - \begin{pmatrix} 3,9106 \\ 1,3056 \end{pmatrix}$$

dont la solution est $\overrightarrow{\delta} x = [-0,5048 \quad 0,10106]^t$. On a maintenant :

$$\begin{array}{rcccccl} x_1^2 & = & x_1^1 + \delta x_1 & = & 2,0211 - 0,50480 & = & 1,5163 \\ x_2^2 & = & x_2^1 + \delta x_2 & = & 3,63604 + 0,10106 & = & 3,7371. \end{array}$$

3. Itération 3 À la troisième itération, on doit résoudre :

$$\begin{pmatrix} 4,5554 & -1 \\ 3,0326 & 7,4742 \end{pmatrix} \begin{pmatrix} \delta x_1 \\ \delta x_2 \end{pmatrix} = - \begin{pmatrix} 0,81824 \\ 0,26508 \end{pmatrix}$$

ce qui entraîne que $\overrightarrow{\delta} x = [-0,17208 \quad 0,034355]^t$. La nouvelle solution est :

$$\begin{array}{rcccccl} x_1^3 & = & x_1^2 + \delta x_1 & = & 1,5163 - 0,17208 & = & 1,3442 \\ x_2^3 & = & x_2 + \delta x_2 & = & 3,7371 + 0,034355 & = & 3,7715. \end{array}$$

4. Itération 4 Le système linéaire à résoudre est :

$$\begin{pmatrix} 3,8351 & -1 \\ 2,6884 & 7,5430 \end{pmatrix} \begin{pmatrix} \delta x_1 \\ \delta x_2 \end{pmatrix} = - \begin{pmatrix} 0,063617 \\ 0,031086 \end{pmatrix}$$

ce qui entraîne que $\overrightarrow{\delta} x = [-0,0161616 \quad 0,0163847]^t$. *La nouvelle approximation de la solution est :*

$$\begin{array}{rcccl} x_1^4 & = & x_1^3 + \delta x_1 & = & 1,3442 - 0,0161616 \quad = \quad 1,3280 \\ x_2^4 & = & x_2^3 + \delta x_2 & = & 3,7715 + 0,0163847 \quad = \quad 3,7731. \end{array}$$

5. Itération 5 *À partir de* $[1,3280 \quad 3,7731]^t$, *on doit résoudre :*

$$\begin{pmatrix} 3,7735 & -1 \\ 2,6560 & 7,5463 \end{pmatrix} \begin{pmatrix} \delta x_1 \\ \delta x_2 \end{pmatrix} = - \begin{pmatrix} 0,34886 \times 10^{-3} \\ 0,16946 \times 10^{-3} \end{pmatrix}$$

dont la solution est $\overrightarrow{\delta} x = [9,03 \times 10^{-5} \quad 9,25 \times 10^{-6}]^t$. *La solution du système non linéaire devient :*

$$\begin{array}{rcccl} x_1^5 & = & x_1^4 + \delta x_1 & = & 1,3281 \\ x_2^5 & = & x_2^4 + \delta x_2 & = & 3,7731. \end{array}$$

On déduit la convergence de l'algorithme de Newton du fait que les modules de $\overrightarrow{\delta x}$ *et de* $\overrightarrow{R}$ *diminuent avec les itérations.*

Remarque 3.6.1.

1. La convergence de la méthode de Newton dépend de l'approximation initiale $\overrightarrow{x}^0$ *de la solution. Un mauvais choix de* $\overrightarrow{x}^0$ *peut résulter en un algorithme divergent.*

2. On peut démontrer que, lorsqu'il y a convergence de l'algorithme, cette convergence est généralement quadratique dans le sens suivant :

$$\|\overrightarrow{x} - \overrightarrow{x}^{i+1}\| \simeq C \|\overrightarrow{x} - \overrightarrow{x}^i\|^2 \tag{3.31}$$

ce qui devient, en posant $\overrightarrow{e}^i = \overrightarrow{x} - \overrightarrow{x}^i$ *:*

$$\|\overrightarrow{e}^{i+1}\| \simeq C \|\overrightarrow{e}^i\|^2.$$

Cela signifie que la norme de l'erreur à l'itération $i+1$ *est approximativement égale à une constante* C *multipliée par le carré de la norme de l'erreur à l'étape* i. *L'analogie est évidente avec le cas d'une seule équation non linéaire étudié au chapitre* 2. *En effet, on peut écrire les deux algorithmes sous la forme :*

$$\begin{array}{rcll} x_{i+1} & = & x_i - (f'(x_i))^{-1} f(x_i) & \text{en dimension } 1 \\ \overrightarrow{x}_{i+1} & = & \overrightarrow{x}_i - (J(\overrightarrow{x}_i))^{-1} \overrightarrow{R}(\overrightarrow{x}_i) & \text{en dimension } n. \end{array}$$

3. La convergence quadratique est perdue si la matrice jacobienne est singulière au point $\overrightarrow{x}$, *solution du système non linéaire. Encore une fois, ce comportement est analogue au cas d'une seule équation où la méthode de Newton perd sa convergence quadratique si la racine est de multiplicité plus grande que* 1 *(car* $f'(r) = 0$*).*

4. Pour obtenir la convergence quadratique, on doit calculer et décomposer une matrice de taille n *sur* n *à chaque itération. De plus, il faut fournir à un éventuel programme informatique les* n *fonctions* $f_i(\overrightarrow{x})$ *et les* n^2 *dérivées partielles de ces fonctions. Cela peut devenir rapidement fastidieux et coûteux lorsque la dimension* n *du système est grande.*

5. Il existe une variante de la méthode de Newton qui évite le calcul des n^2 *dérivées partielles et qui ne nécessite que les* n *fonctions* $f_i(\overrightarrow{x})$. ***La méthode de Newton modifiée*** *consiste à remplacer les dérivées partielles par des différences centrées (voir le chapitre). On utilise alors l'approximation du second ordre* $(O(h^2))$ *:*

$$\frac{\partial f_i}{\partial x_j}(x_1, x_2, ..., x_n) \simeq \frac{f_i(x_1, ..., x_{j-1}, x_j + h, ..., x_n) - f_i(x_1, ..., x_{j-1}, x_j - h, ..., x_n)}{2h} \tag{3.32}$$

Cette approximation introduit une petite erreur dans le calcul de la matrice jacobienne, mais généralement la convergence est quand même très rapide.

Exemple 3.6.2. *Dans l'exemple précédent, le calcul du premier terme de la matrice jacobienne de la première itération donnait :*

$$\frac{\partial f_1}{\partial x_1}(2,8\ ;\ 2,8) = e^{2,8} = 16,44464677$$

tandis que l'approximation 3.32 donne pour h = 0,001 :

$$\frac{f_1(2,801\ ;\ 2,8) - f_1(2,799\ ;\ 2,8)}{(2)(0,001)} = 16,44465.$$

On constate que l'erreur introduite est minime.

Chapitre 4

Équations différentielles

4.1 Introduction

La résolution numérique des équations différentielles est probablement le domaine de l'analyse numérique où les applications sont les plus nombreuses. Que ce soit en mécanique des fluides, en transfert de chaleur ou en analyse de structures, on aboutit souvent à la résolution d'équations différentielles, de systèmes d'équations différentielles ou plus généralement d'équations aux dérivées partielles.

Nous prenons comme point de départ la formulation générale d'une équa-tion différentielle d'ordre 1 avec condition initiale. La tâche consiste à déter-miner une fonction y(t) solution de :

$$\begin{cases} y'(t) = f(t, y(t)) \\ \\ y(t_0) = y_0. \end{cases} \tag{4.1}$$

La variable indépendante t représente très souvent (mais pas toujours) le temps. La variable dépendante est notée y et dépend bien sûr de t. La fonction f est pour le moment une fonction quelconque de deux variables que nous supposons suffisamment différentiable. La condition $y(t_0) = y_0$ est la condition initiale et en quelque sorte l'état de la solution au moment où l'on commence à s'y intéresser. Il s'agit d'obtenir $y(t)$ pour $t = t_0$, si l'on cherche une solution analytique, ou une approximation de $y(t)$, si l'on utilise une méthode numérique.

Définition 4.1.1. *L'équation différentielle 4.1 est dite d'ordre* 1, *car seule la dérivée d'ordre* 1 *de la variable dépendante* $y(t)$ *est présente. Si des dérivées de* $y(t)$ *d'ordre* 2 *apparaissaient dans l'équation différentielle 4.1, on aurait une équation d'ordre* 2, *et ainsi de suite.*

4.2 Méthode d'Euler explicite

La méthode d'Euler explicite est de loin la méthode la plus simple de résolution numérique d'équations différentielles ordinaires. Elle possède une belle interprétation géométrique et son emploi est facile. Toutefois, elle est relativement peu utilisée en raison de sa faible précision. On la qualifie d'explicite car elle ne nécessite pas de résolution d'équation non linéaire contrairement à la méthode d'Euler dite implicite que nous verrons plus loin. Reprenons l'équation différentielle 4.1 et considérons plus attentivement la condition initiale $y(t_0) = y_0$. Le but est maintenant d'obtenir une approximation de la solution en $t = t_1 = t_0 + h$. Avant d'effectuer la première itération, il faut déterminer dans quelle direction on doit avancer à partir du point (t_0, y_0) pour obtenir le point (t_1, y_1), qui est une approximation du point $(t_1, y(t_1))$. Nous n'avons pas l'équation de la courbe $y(t)$, mais nous en connaissons la pente $y'(t)$ en $t = t_0$. En effet, l'équation différentielle assure que :

$$y'(t_0) = f(t_0, y(t_0)) = f(t_0, y_0).$$

On peut donc suivre la droite passant par (t_0, y_0) et de pente $f(t_0, y_0)$. L'équation de cette droite, notée $d_0(t)$, est : $d_0(t) = y_0 + f(t_0, y_0)(t - t_0)$.

En $t = t_1$, on a : $d_0(t_1) = y_0 + f(t_0, y_0)(t_1 - t_0) = y_0 + hf(t_0, y_0) = y_1$

En d'autres termes, $d_0(t_1)$ est proche de la solution analytique $y(t_1)$, c'est-à-dire : $y(t_1) \simeq y_1 = d_0(t_1) = y_0 + hf(t_0, y_0)$. Il est important de noter que, le plus souvent, $y_1 = y(t_1)$. Cette inégalité n'a rien pour étonner, mais elle a des conséquences sur la suite du raisonnement. En effet, si l'on souhaite faire une deuxième itération et obtenir une approximation de $y(t_2)$, on peut refaire l'analyse précédente à partir du point (t_1, y_1). On remarque cependant que la pente de la solution analytique en $t = t_1$ est : $y'(t_1) = f(t_1, y(t_1))$. On ne connaît pas exactement $y(t_1)$, mais on possède l'approximation y_1 de $y(t_1)$. On doit alors utiliser l'expression : $y'(t_1) = f(t_1, y(t_1)) \simeq f(t_1, y_1)$ et construire la droite : $d_1(t) = y_1 + f(t_1, y_1)(t - t_1)$ qui permettra d'estimer $y(t_2)$. On constate que l'erreur commise à la première itération est réintroduite dans les calculs de la deuxième itération. On a alors : $y(t_2) \simeq y_2 = d_1(t_2) = y_1 + hf(t_1, y_1)$.

Remarque 4.2.1. *Le développement qui précède met en évidence une propriété importante des méthodes numériques de résolution des équations différentielles. En effet, l'erreur introduite à la première itération a des répercussions sur les calculs de la deuxième itération, ce qui signifie que les erreurs se propagent d'une itération à l'autre. Il en résulte de façon générale que l'erreur :* $|y(t_n) - y_n|$ *augmente légèrement avec* n.

On en arrive donc à l'algorithme suivant.

Algorithme : Méthode d'Euler explicite

1. Étant donné un pas de temps h, une condition initiale (t_0, y_0) et un nombre maximal d'itérations N.

2. Pour $0 \leq n \leq N$:
$y_{n+1} = y_n + hf(t_n, y_n)$
$t_{n+1} = t_n + h$
Écrire t_{n+1} et y_{n+1}

3. Arrêt

Exemple 4.2.1. *Soit l'équation différentielle :* $y'(t) = -y(t)+t+1$ *et la condition initiale* $y(0) = 1$. *On a donc* $t_0 = 0$ *et* $y_0 = 1$ *et l'on prend un pas de temps* $h = 0{,}1$. *De plus, on a :* $f(t, y) = -y+t+1$ *On peut donc utiliser la méthode d'Euler explicite et obtenir successivement des approximations de* $y(0,1), y(0,2), y(0,3), ...,$ *notées* $y_1, y_2, y_3....$ *Le premier pas de temps produit :*

$$y_1 = y_0 + hf(t_0, y_0) = 1 + 0,1f(0,1) = 1 + 0,1(-1 + 0 + 1) = 1$$

Le deuxième pas de temps fonctionne de manière similaire :

$$y_2 = y_1 + hf(t_1, y_1) = 1 + 0,1f(0,1,1) = 1 + 0,1(-1 + 0,1 + 1) = 1,01.$$

On parvient ensuite à :

$$y_3 = y_2 + hf(t_2, y_2) = 1,01 + 0,1f(0,2,1,01) = 1,01 + 0,1(-1,01 + 0,2 + 1) = 1,029.$$

Le tableau suivant rassemble les résultats des dix premiers pas de temps. La solution analytique de cette équation différentielle est $y(t) = e^{-t} + t$, *ce qui permet de comparer les solutions numérique et analytique et de constater la croissance de l'erreur.*

Méthode d'Euler explicite : $\mathbf{y'(t) = y(t) + t + 1}$

$\mathbf{t_i}$	$\mathbf{y(t_i)}$	$\mathbf{y_i}$	$\mathbf{\lvert y(t_i) - y_i \rvert}$
$0,0$	$1,000000$	$1,000000$	$0,000000$
$0,1$	$1,004837$	$1,000000$	$0,004837$
$0,2$	$1,018731$	$1,010000$	$0,008731$
$0,3$	$1,040818$	$1,029000$	$0,011818$
$0,4$	$1,070302$	$1,056100$	$0,014220$
$0,5$	$1,106531$	$1,090490$	$0,016041$
$0,6$	$1,148812$	$1,131441$	$0,017371$
$0,7$	$1,196585$	$1,178297$	$0,018288$
$0,8$	$1,249329$	$1,230467$	$0,018862$
$0,9$	$1,306570$	$1,287420$	$0,019150$
$1,0$	$1,367879$	$1,348678$	$0,019201$

Définition 4.2.1. *Une méthode de résolution d'équations différentielles est dite à un pas si elle est de la forme :*

$$y_{n+1} = y_n + h\phi(t_n, y_n) \tag{4.2}$$

où ϕ est une fonction quelconque. Une telle relation est appelée équation aux différences. La méthode est à un pas si, pour obtenir la solution en $t = t_{n+1}$, on doit utiliser la solution numérique au temps t_n seulement. On désigne méthodes à pas multiples les méthodes qui exigent également la solution nu-mérique aux temps $t_{n-1}, t_{n-2}, t_{n-3}, ...$

Définition 4.2.2. *On dira qu'un schéma à un pas converge à l'ordre p si :*

$$\max_{1 \leq n \leq N} |y(t_n) - y_n| = O(h^p) \tag{4.3}$$

où N est le nombre total de pas de temps. L'ordre de convergence d'une méthode à un pas dépend de l'erreur commise à chaque pas de temps via l'erreur de troncature locale que nous allons maintenant définir.

Définition 4.2.3. *L'erreur de troncature locale au point $t = t_n$ est définie par :*

$$t_{n+1}(h) = y(t_{n+1}) - y(t_n)h - \phi(t_n, y(t_n)) \tag{4.4}$$

L'erreur de troncature locale mesure la précision avec laquelle la solution analytique vérifie l'équation aux différences 4.2.

Examinons plus avant le cas de la méthode d'Euler explicite ($\phi(t, y) = f(t, y)$). Ici encore, l'outil de travail est le développement de Taylor. En effectuant un développement autour du point $t = t_n$, on trouve :

$$y(t_{n+1}) = y(t_n + h) = y(t_n) + y'(t_n)h + \frac{y''(t_n)h^2}{2} + O(h^3) = y(t_n) + f(t_n, y(t_n))h + \frac{y''(t_n)h^2}{2} + O(h^3)$$

puisque $y'(t_n) = f(t_n, y(t_n))$. L'erreur de troncature locale 4.4 devient donc :

$$t_{n+1}(h) = y(t_{n+1}) - y(t_n)h - f(t_n, y(t_n)) = y''(t_n)h^2 + O(h^2)$$

ou plus simplement $t_{n+1}(h) = O(h)$ et la méthode d'Euler explicite converge donc à l'ordre 1 ($p = 1$ dans la relation 4.3).

4.3 Méthodes de Taylor

Le développement de Taylor autorise une généralisation immédiate de la méthode d'Euler, qui permet d'obtenir des algorithmes dont l'erreur de tronca-ture locale est d'ordre plus élevé. Nous nous limitons cependant à la méthode de Taylor du second ordre. On cherche, au temps $t = t_n$, une approximation de la solution en $t = t_{n+1}$. On a immédiatement :

$$y(t_{n+1}) = y(t_n + h) = y(t_n) + y'(t_n)h + \frac{y''(t_n)h^2}{2} + O(h^3).$$

En se servant de l'équation différentielle 4.1, on trouve :

$$y(t_{n+1}) = y(t_n) + f(t_n, y(t_n))h + \frac{f'(t_n, y(t_n))h^2}{2} + O(h^3).$$

Dans la relation précédente, on voit apparaître la dérivée de la fonction $f(t, y(t))$ par rapport au temps. La règle de dérivation en chaîne assure que :

$$f'(t, y(t)) = \frac{\partial f(t, y(t))}{\partial t} + \frac{\partial f(t, y(t))}{\partial y} y'(t)$$

c'est-à-dire :

$$f'(t, y(t)) = \frac{\partial f(t, y(t))}{\partial t} + \frac{\partial f(t, y(t))}{\partial y} f(t, y(t)).$$

On obtient donc :

$$y(t_{n+1}) = y(t_n) + hf(t_n, y(t_n)) + \frac{h^2}{2}\left(\frac{\partial f(t_n, y(t_n))}{\partial t} + \frac{\partial f(t_n, y(t_n))}{\partial y} f(t_n, y(t_n))\right) + O(h^3) \quad (4.5)$$

En négligeant les termes d'ordre supérieur ou égal à 3, on en arrive à poser :

$$y(t_{n+1}) \simeq y(t_n) + hf(t_n, y(t_n)) + \frac{h^2}{2}(\frac{\partial f(t_n, y(t_n))}{\partial t} + \frac{\partial f(t_n, y(t_n))}{\partial y} f(t_n, y(t_n))) \quad (4.6)$$

qui sera à la base de la méthode de Taylor.

Remarque 4.3.1. *Il importe de préciser l'ordre de troncature locale de cette méthode. Dans ce cas, suivant la notation 4.3, on a :*

$$\phi(t, y(t)) = f(t, y(t)) + \frac{h}{2}\left(\frac{\partial f(t, y(t))}{\partial t} + \frac{\partial f(t, y(t))}{\partial y} f(t, y(t))\right).$$

En vertu de la relation 4.5 et de la définition de l'erreur de troncature locale 4.4, il est facile de montrer que : $t_{n+1}(h) = O(h^2)$. L'erreur de troncature locale de la méthode de Taylor est d'ordre 2 et la méthode converge à l'ordre 2 ($p = 2$ dans la relation 4.3).

En remplaçant $y(t_n)$ par y_n dans l'équation 4.6, on arrive à l'algorithme de la méthode de Taylor d'ordre 2.

Algorithme : Méthode de Taylor d'ordre 2

1. Étant donné un pas de temps $\mathbf{h}$, une condition initiale $(\mathbf{t_0}, \mathbf{y_0})$ et un nombre maximal d'itérations $\mathbf{N}$.

2. Pour $\mathbf{0 \leq n \leq N}$:
$\mathbf{y_{n+1}} = \mathbf{y_n} + \mathbf{hf}(\mathbf{t_n}, \mathbf{y_n}) + \dfrac{\mathbf{h^2}}{\mathbf{2}}\left(\dfrac{\partial \mathbf{f}(\mathbf{t_n}, \mathbf{y_n})}{\partial \mathbf{t}} + \dfrac{\partial \mathbf{f}(\mathbf{t_n}, \mathbf{y_n})}{\partial \mathbf{y}} \mathbf{f}(\mathbf{t_n}, \mathbf{y_n})\right)$
$\mathbf{t_{n+1}} = \mathbf{t_n} + \mathbf{h}$
Écrire $\mathbf{t_{n+1}}$ et $\mathbf{y_{n+1}}$.

3. Arrêt $\mathbf{N}$.

Exemple 4.3.1. *Soit l'équation différentielle déjà résolue par la méthode d'Euler :* $y'(t) = -y(t) + t + 1$ *et la condition initiale* $y(0) = 1$. *Dans ce cas :* $f(t, y) = -y + t + 1$ *de même que* $\dfrac{\partial f}{\partial t} = 1$ *et* $\dfrac{\partial f}{\partial y} = -1$. *L'algorithme devient :*

$$y_{n+1} = y_n + h(-y_n + t_n + 1) + \frac{h^2}{2}\left(1 + (-1)(-y_n + t_n + 1)\right).$$

La première itération de la méthode de Taylor d'ordre 2 donne (avec $h = 0,1$*) :* $y_1 = 1 + 0,1(-1 + 0 + 1) + (0,1)22(1 + (-1)(-1 + 0 + 1)) = 1,005$. *Une deuxième itération donne :* $y_2 = 1,005 + 0,1(-1,005 + 0,1 + 1) + (0,1)22(1 + (-1)(-1,005 + 0,1 + 1)) = 1,019025$. *Les résultats sont compilés dans le tableau qui suit.*

Méthode de Taylor *:* $\mathbf{y'(t) = y(t) + t + 1}$

$\mathbf{t_i}$	$\mathbf{y(t_i)}$	$\mathbf{y_i}$	$\mathbf{\lvert y(t_i) - y_i \rvert}$
$0,0$	$1,000000$	$1,000000$	$0,000000$
$0,1$	$1,004837$	$1,005000$	$0,000163$
$0,2$	$1,018731$	$1,019025$	$0,000294$
$0,3$	$1,040818$	$1,041218$	$0,000400$
$0,4$	$1,070302$	$1,070802$	$0,000482$
$0,5$	$1,106531$	$1,107075$	$0,000544$
$0,6$	$1,148812$	$1,149404$	$0,000592$
$0,7$	$1,196585$	$1,197210$	$0,000625$
$0,8$	$1,249329$	$1,249975$	$0,000646$
$0,9$	$1,306570$	$1,307228$	$0,000658$
$1,0$	$1,367879$	$1,368541$	$0,000662$

On remarque que l'erreur est plus petite avec la méthode de Taylor d'ordre 2 *qu'avec la méthode d'Euler explicite. Comme on le verra plus loin, cet avantage des méthodes d'ordre plus élevé vaut pour l'ensemble des méthodes de résolution d'équations différentielles.*

Exemple 4.3.2. *Soit l'équation différentielle :*

$$\begin{cases} y'(t) = ty(t) \\ y(1) = 2 \end{cases} \tag{4.7}$$

Convergence de la méthode de Taylor : $\mathbf{y'(t) = ty(t)}$

h	i	y_i	$\lvert y(t) - y_i \rvert$	Rapport
$0,5000000$	2	$7,546875$	$1,41700$	$-$
$0,2500000$	4	$8,444292$	$0,51900$	$2,72$
$0,1250000$	8	$8,804926$	$0,15800$	$3,27$
$0,0625000$	16	$8,919646$	$0,04370$	$3,61$
$0,0312500$	32	$8,951901$	$0,01140$	$3,80$
$0,0156250$	64	$8,960439$	$0,00290$	$3,87$
$0,0078125$	128	$8,962635$	$0,00074$	$3,90$

4.4 Méthodes de Runge-Kutta

Il serait avantageux de disposer de méthodes d'ordre de plus en plus élevé tout en évitant les désavantages des méthodes de Taylor, qui nécessitent l'évaluation des dérivées partielles de la

fonction $f(t, y)$. Une voie est tracée par les méthodes de Runge-Kutta, qui sont calquées sur les méthodes de Taylor du même ordre.

4.4.1 Méthodes de Runge-Kutta d'ordre 2

On a vu que le développement de la méthode de Taylor passe par la relation 4.7 :

$$y(t_{n+1}) = y(t_n) + hf(t_n, y(t_n)) + \frac{h^2}{2}\left(\frac{\partial f(t_n, y(t_n))}{\partial t} + \frac{\partial f(t_n, y(t_n))}{\partial y} f(t_n, y(t_n))\right) + O(h^3) \quad (4.8)$$

Le but est de remplacer cette dernière relation par une expression équivalente possédant le même ordre de précision ($O(h^3)$). On propose la forme :

$$y(t_{n+1}) = y(t_n) + a_1 hf(t_n, y(t_n)) + a_2 hf(t_n + a_3 h, y(t_n) + a_4 h) \quad (4.9)$$

où l'on doit déterminer les paramètres a_1, a_2, a_3 et a_4 de telle sorte que les expressions 4.8 et 4.9 aient toutes deux une erreur en $O(h^3)$. On ne trouve par ailleurs aucune dérivée partielle dans cette expression. Pour y arriver, on doit recourir au développement de Taylor en deux variables autour du point $(t_n, y(t_n))$. On a ainsi :

$$f(t_n + a_3 h, y(t_n) + a_4 h) = f(t_n, y(t_n)) + a_3 h \frac{\partial f(t_n, y(t_n))}{\partial t} + a_4 h \frac{\partial f(t_n, y(t_n))}{\partial y} + O(h^2).$$

La relation 4.9 devient alors :

$$y(t_{n+1}) = y(t_n) + (a_1 + a_2) hf(t_n, y(t_n)) + a_2 a_3 h^2 \frac{\partial f(t_n, y(t_n))}{\partial t} + a_2 a_4 h^2 \frac{\partial f(t_n, y(t_n))}{\partial y} + O(h^3). \quad (4.10)$$

On voit immédiatement que les expressions 4.8 et 4.10 sont du même ordre. Pour déterminer les coefficients a_i, il suffit de comparer ces deux expressions terme à terme :

- coefficients respectifs de $f(tn, y(tn)) : h = (a1 + a2)h$
- coefficients respectifs de $?f(tn, y(tn))?t : h22 = a2a3h2$
- coefficients respectifs de $?f(tn, y(tn))?y : h22f(tn, y(tn)) = a2a4h2$. On obtient ainsi un système non linéaire de 3 équations comprenant 4 inconnues :

$$\begin{cases} 1 = (a_1 + a_2) \\ \dfrac{1}{2} = a_2 a_3 \\ \dfrac{f(t_n, y(t_n))}{2} = a_2 a_4 \end{cases} \quad (4.11)$$

Le système 4.11 est sous-déterminé en ce sens qu'il y a moins d'équations que d'inconnues et qu'il n'a donc pas de solution unique. Cela offre une marge de manoeuvre qui favorise la mise au point de plusieurs variantes de la méthode de Runge-Kutta. Voici le choix le plus couramment utilisé. Méthode d'Euler modifiée $a_1 = a_2 = \frac{1}{2}, a_3 = 1$ et $a_4 = f(t_n, y(t_n))$. On établit sans peine que ces coefficients satisfont aux trois équations du système non linéaire 4.11. Il suffit ensuite de remplacer ces valeurs dans l'équation 4.9. Pour ce faire, on doit négliger le terme en $O(h^3)$ et remplacer la valeur exacte $y(t_n)$ par son approximation y_n. On obtient alors l'algorithme suivant.

Algorithme : Méthode d'Euler modifiée

1. Étant donné un pas de temps h, une condition initiale (t_0, y_0) et un nombre maximal d'itérations N

2. Pour $0 \leq n \leq N$:

$\widehat{y} = y_n + hf(t_n, y_n)$
$y_{n+1} = y_n + \dfrac{h}{2}(f(t_n, y_n) + f(t_n + h, \widehat{y}))$
$t_{n+1} = t_n + h$
Écrire t_{n+1} et y_{n+1}

3. Arrêt N.