
ANALYSE NUMERIQUE 1

SALEM NAFIRI¹

Ecole Hassania des Travaux Publics 2018-2019

1. Email : nafiri.ehtp@gmail.com

Table des matières

Introduction	5
1 Interpolation polynomiale	7
1.1 Motivations et exemple	7
1.2 Position du problème	8
1.3 Base de Lagrange	9
1.4 Interpolation de Lagrange	10
1.5 Interpolation d'une fonction continue par un polynôme	11
1.6 Interpolation d'Hermite	12
2 Résolution numérique d'équations nonlinéaires	15
2.1 Position du problème	15
2.2 Méthode de dichotomie ou de bisection	15
2.3 Méthode de Newton-Raphston	17
2.4 Théorème du point fixe et méthode d'approximations successives	18
3 Résolution d'un système linéaire	21
3.1 Position du problème	21
3.2 Méthode de Gauss sur un exemple	22
3.2.1 Algorithme d'élimination de Gauss	23
3.2.2 Compte des opérations élémentaires dans la méthode de Gauss	25
3.3 Décomposition LU d'une matrice	26
3.4 Décomposition de Cholesky	28

Introduction

C'est quoi l'analyse numérique ?

D'après *Wikipédia*, l'analyse numérique est une discipline des mathématiques qui s'intéresse aux fondements théoriques et à la mise en pratique des méthodes permettant de résoudre (par des calculs purement numériques) des problèmes d'analyse mathématique.

Selon *Nick Trefethen* (spécialiste en analyse numérique à l'université d'Oxford en Angleterre), l'analyse numérique est l'étude des algorithmes pour les problèmes des mathématiques continues.

Tout simplement l'analyse numérique est l'art de passer des mathématiques continues aux mathématiques discrètes.

Pourquoi faire de l'analyse numérique ?

L'objet de l'analyse numérique est la conception et l'étude de méthodes de résolution numérique (on dit aussi approximation numérique) des modèles mathématiques des sciences de l'ingénieur, des sciences expérimentales, de l'économie etc La solution exacte du modèle mathématique étant impossible à calculer explicitement. Il est souvent plus intéressant d'approcher la solution d'un modèle compliqué réaliste que de calculer exactement la solution d'un modèle simplifié mais loin de la réalité. L'outil de base qu'utilise l'analyse numérique est l'ordinateur (calculateur = Computer). A partir de la méthode d'approximation, nous écrivons un algorithme puis un programme informatique que nous mettons en œuvre sur la machine (on dit aussi qu'on l'implémente sur machine). L'efficacité d'une méthode dépend aussi bien de ses propriétés mathématiques (convergence, précision, etc...) que de sa facilité d'implémentation et de son comportement sur la machine. Dans ce cours nous étudierons quelques méthodes de base de l'analyse numérique.

Ce cours sera complété par des travaux pratiques sous l'environnement de calcul scientifique MATLAB.

Quelles sont les applications de l'analyse numérique ?

Les champs d'applications de l'analyse numérique sont diverses et variés suivant le problème à résoudre :

- Médecine : Modélisation de la dynamique de Sida, contrôle numérique du Malarya, ...

- Optimisation des formes : conception des avions, des voitures, des ordinateurs, ...
- Météorologie et modèles climatiques : prévision numérique du temps
- Génie civil : conception des multistruktures comme les ponts, les ouvrages d'art (Tour Eiffel, Statue de liberté, ...)

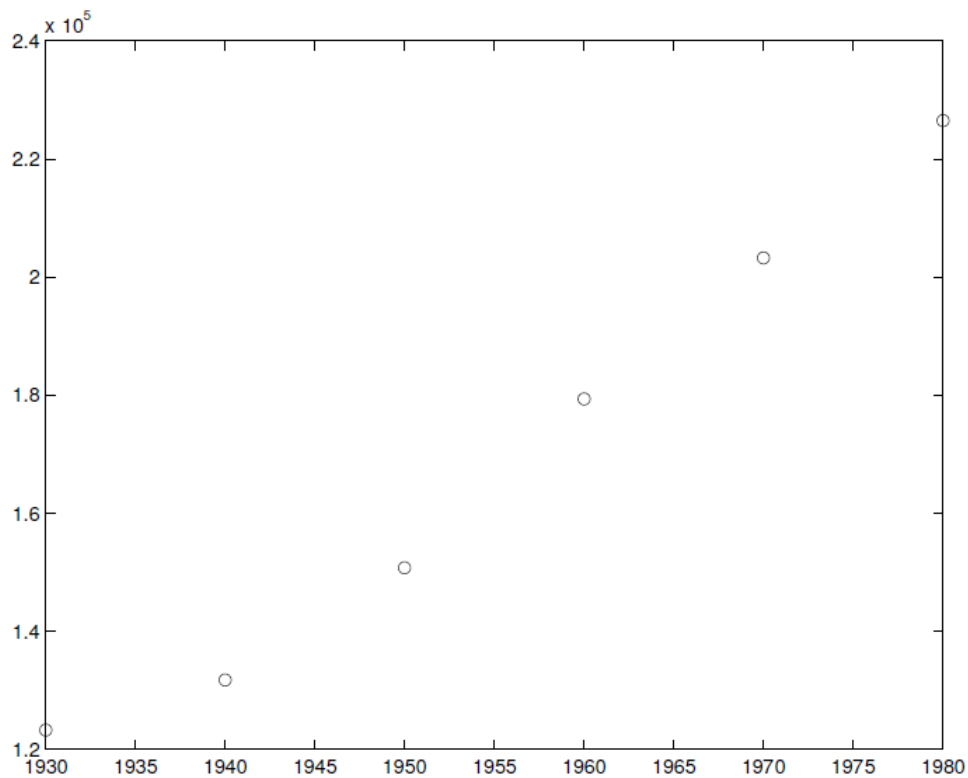
Chapitre 1

Interpolation polynomiale

1.1 Motivations et exemple

Un pays fait un recensement de sa population tous les dix ans par exemple :

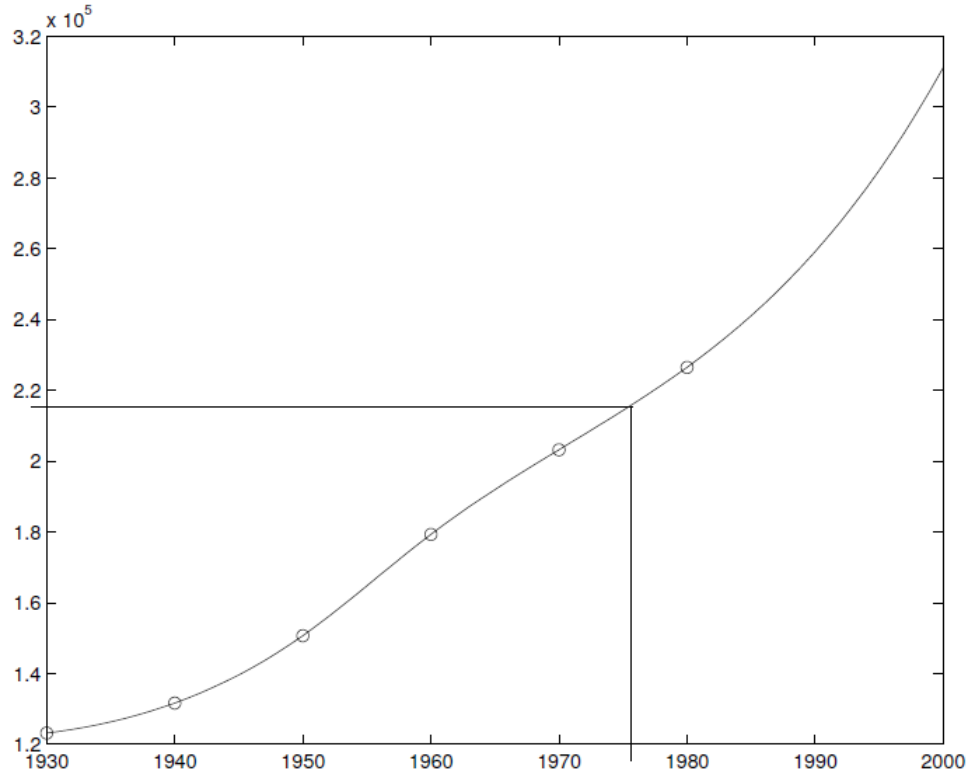
année t_i	1930	1940	1950	1960	1970	1980
Population $p_i \times 1000$	123203	131669	150697	179323	203212	226505



Parfois nous avons besoins d'estimer la population en 1975 ou en 2000. Ce type de prédiction peut être obtenu à partir d'une fonction $f(\cdot)$ qui coïncide avec les données disponibles ou les mesures effectuées, c'est à dire telle que

$$f(t_i) = p_i, \text{ pour } i = 1, \dots, n.$$

On dit que f interpole les données $(t_i, p_i)_{i=1}^n$. Par exemple :



La recherche de f peut se faire dans plusieurs classes de fonctions. Une classe très simple est celle des polynômes, facile à dériver, à intégrer et de plus nous avons le théorème d'approximation de Weistrass :

Théorème 1.1.1. Soit f une fonction continue sur $[a, b]$ et $\epsilon > 0$. Alors il existe un polynôme p tel que

$$\forall x \in [a, b] \quad |f(x) - p(x)| \leq \epsilon.$$

1.2 Position du problème

Supposons que l'on veuille chercher un polynôme p de degré $n \geq 0$ qui, pour des valeurs $t_0, t_1, t_2, \dots, t_n$ distinctes données, prenne les valeurs $p_0, p_1, p_2, \dots, p_n$ respectivement, c'est à dire

$$p(t_j) = p_j \quad \text{pour } 0 \leq j \leq n. \quad (1.1)$$

Une manière simple de résoudre ce problème est d'écrire

$$p(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + \cdots + a_nt^n$$

où a_0, a_1, \dots, a_n sont des coefficients qui devront être déterminés (clairement si les coefficients a_j , $0 \leq j \leq n$ sont connus alors le polynôme p est connu.) Les $(n+1)$ relations (1.1) s'écrivent alors

$$a_0 + a_1t_j + a_2t_j^2 + a_3t_j^3 + \cdots + a_nt_j^n = p_j, \quad 0 \leq j \leq n.$$

Puisque les valeurs t_j et p_j , $0 \leq j \leq n$, sont connues, les relations précédentes forment un système de $(n+1)$ équations à $(n+1)$ inconnues a_0, a_1, \dots, a_n . Matriciellement parlant, les relations précédentes peuvent s'écrire

$$T\vec{a} = \vec{p}$$

où

$$T = \begin{pmatrix} 1 & t_0 & t_0^2 & \cdots & t_0^n \\ 1 & t_1 & t_1^2 & \cdots & t_1^n \\ 1 & t_2 & t_2^2 & \cdots & t_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_n & t_n^2 & \cdots & t_n^n \end{pmatrix}, \quad \vec{a} = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}, \quad \vec{p} = \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \\ p_n \end{pmatrix}.$$

Définition 1.2.1. Nous dirons que T est la matrice de Vandermonde associée aux points $t_0, t_1, t_2, \dots, t_n$.

Ainsi, le problème consistant à chercher le polynôme p satisfaisant (1.1) peut se réduire à résoudre le système linéaire $Ta = p$, c'est à dire à déterminer a puisque T et p sont connus.

Remarque 1.2.1. Résoudre un système linéaire de $(n+1)$ équations à $(n+1)$ inconnues n'est pas une tâche triviale (voir chap.4). La méthode que nous venons de décrire pour trouver le polynôme p n'est pas une bonne méthode. Puisque le nombre d'opérations arithmétiques nécessaires pour inverser T (par Jordan-Gauss) prend approximativement $\frac{2}{3}n^3$ opérations, c'est à dire $O(n^3)$ (si la matrice est triangulaire, ce nombre est divisé par 2, soit $\frac{n^3}{3}$ opérations.) Dans la suite nous présentons une méthode plus astucieuse pour construire le polynôme p .

1.3 Base de Lagrange

Il est facile de résoudre le problème (1.1) lorsque toutes les valeurs p_j sont égales à zéro sauf une, qui est fixée à 1. Soit k un entier donnée entre 0 et n et supposons que l'on ait $p_k = 1$ et $p_j = 0$ pour $j \neq k$. Soit φ_k la fonction de t définie par

$$\varphi_k(t) = \frac{(t-t_0)(t-t_1)\cdots(t-t_{k-1})(t-t_{k+1})\cdots(t-t_n)}{(t_k-t_0)(t_k-t_1)\cdots(t_k-t_{k-1})(t_k-t_{k+1})\cdots(t_k-t_n)}.$$

Clairement le numérateur de φ_k est un produit de n termes $(t-t_j)$ avec $j \neq k$ et est donc un polynôme de degré n en t . Le dénominateur de φ_k est une constante et il est alors facile de vérifier que

- (i) φ_k est un polynôme de degré n ,
- (ii) $\varphi_k(t_j) = 0$ si $j \neq k, 0 \leq j \leq n$,
- (iii) $\varphi_k(t_k) = 1$.

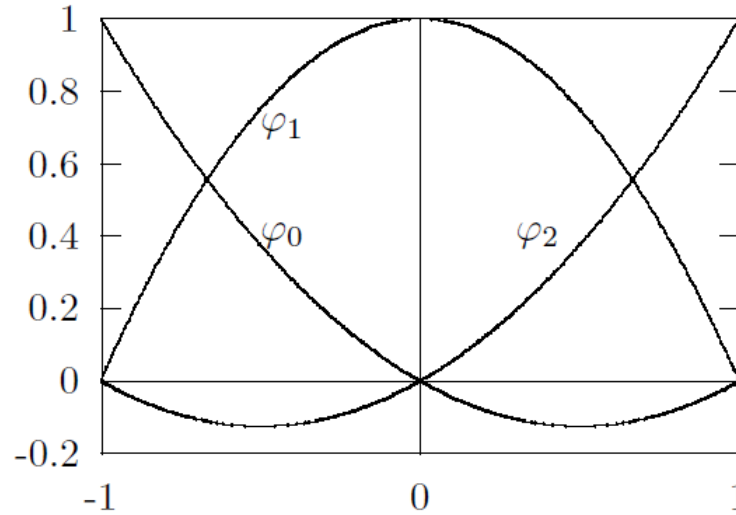
Notons par \mathbb{P}_n l'espace vectoriel formé par tous les polynômes de degré inférieur ou égal à n . Il est bien connu que \mathbb{P}_n est de dimension $(n+1)$ et que sa base canonique est donnée par $\{1, t, t^2, \dots, t^n\}$. On peut montrer que $\varphi_0, \varphi_1, \varphi_2, \dots, \varphi_n$ sont linéairement indépendants, donc forment aussi une base de \mathbb{P}_n . Ainsi nous adoptons la définition suivante

Définition 1.3.1. Nous dirons que $\varphi_0, \varphi_1, \varphi_2, \dots, \varphi_n$ est la base de Lagrange de \mathbb{P}_n associés aux points $t_0, t_1, t_2, \dots, t_n$.

Exemple 1.3.1. Prenons $n = 2, t_0 = -1, t_1 = 0, t_2 = 1$. La base de Lagrange de \mathbb{P}_2 associée aux points $-1, 0$ et 1 est formée par les polynômes $\varphi_0, \varphi_1, \varphi_2$ définis par

$$\begin{aligned}\varphi_0(t) &= \frac{1}{2}t^2 - \frac{1}{2}t; \\ \varphi_1(t) &= 1 - t^2; \\ \varphi_2(t) &= \frac{1}{2}t^2 + \frac{1}{2}t;\end{aligned}$$

Les graphes de $\varphi_0, \varphi_1, \varphi_2$ sur l'intervalle $[-1, 1]$ sont représentés dans la figure suivante



1.4 Interpolation de Lagrange

Revenons au problème (1.1) consistant à chercher un polynôme p de degré n qui prenne des valeurs données $p_0, p_1, p_2, \dots, p_n$ en des points distincts donnés

$t_0, t_1, t_2, \dots, t_n$. Soit $\varphi_0, \varphi_1, \varphi_2, \dots, \varphi_n$ est la base de Lagrange de \mathbb{P}_n associés aux points $t_0, t_1, t_2, \dots, t_n$. Alors le polynôme p cherché est défini par :

$$p(t) = p_0\varphi_0(t) + p_1\varphi_1(t) + \dots + p_n\varphi_n(t) = \sum_{j=0}^n p_j\varphi_j(t).$$

Remarque 1.4.1. Il est important de remarquer que nous avons construit explicitement une solution du problème (1.1) et ceci pour n'importe quelles valeurs p_0, p_1, \dots, p_n données. Ceci montre que le système $Ta = p$ a toujours une solution a pour n'importe quel p et ainsi la matrice de Vandermonde T est régulière. La solution $p(t) = \sum_{j=0}^n p_j\varphi_j(t)$ du problème (1.1) est donc unique.

Exemple 1.4.1. Trouver un polynôme de degré 2 qui en $t_0 = -1$ vaut $p_0 = 8$, en $t_1 = 0$ vaut $p_1 = 3$ et en $t_2 = 1$ vaut $p_2 = 6$.

D'après l'exemple précédent nous avons $p(t) = 8\varphi_0(t) + 3\varphi_1(t) + 6\varphi_2(t)$. Ce qui entraîne

$$p(t) = 4t^2 - t + 3.$$

1.5 Interpolation d'une fonction continue par un polynôme

Soit $f : [a, b] \longrightarrow \mathbb{R}$ continue donnée et soit $t_0, t_1, t_2, \dots, t_n$, $(n+1)$ points distincts donnés.

Nous cherchons maintenant à interpoler f par un polynôme de degré n aux points $t_j, 0 \leq j \leq n$, c'est à dire nous cherchons un polynôme p de degré n tel que

$$p(t_j) = f(t_j), \quad 0 \leq j \leq n. \quad (1.2)$$

Si f est donnée, alors en posant $p_j = f(t_j)$, $0 \leq j \leq n$ et en suivant ce qui est fait dans la section précédente, nous obtenons $p(t) = \sum_{j=0}^n p_j\varphi_j(t)$.

La solution du problème (1.2) est définie par :

$$p(t) = \sum_{j=0}^n f(t_j)\varphi_j(t) \quad \forall t \in [a, b]. \quad (1.3)$$

Définition 1.5.1. On dira que le polynôme p est l'interpolant de f de degré n aux points $t_0, t_1, t_2, \dots, t_n$.

Exercice 1.5.1. Trouver l'interpolant de $f(t) = e^t$ de degré 2 aux points $-1, 0$ et 1 . Puis tracer f et son interpolant sur le même intervalle.

Soit (t_j) une subdivision de $[a, b]$, $t_j = a + jh$, $0 \leq j \leq n$, avec $h = \frac{b-a}{n}$. Soit p_n l'interpolant de f de degré n aux points $t_0, t_1, t_2, \dots, t_n$. D'après (1.3), p_n est défini par :

$$p_n(t) = \sum_{j=0}^n f(t_j)\varphi_j(t).$$

On peut montrer le résultat suivant

Théorème 1.5.1. *Supposons que $f \in \mathcal{C}^{n+1}([a, b])$. Alors*

$$\max_{t \in [a, b]} |f(t) - p_n(t)| \leq \frac{1}{2(n+1)} \left(\frac{b-a}{n} \right)^{n+1} \max_{t \in [a, b]} |f^{(n+1)}(t)|$$

$$\text{où } f^{(n+1)}(t) = \frac{d^{n+1}f(t)}{dt^{n+1}}.$$

Remarque 1.5.1. *L'inégalité précédente est une estimation d'erreur entre la fonction f et son interpolant p_n . Il se peut que $\max_{t \in [a, b]} |f^{(n+1)}(t)|$ croît très rapidement avec n et donc la convergence de p_n vers f n'aura pas lieu. Ce phénomène s'appelle **phénomène de Runge**.*

Cas particulier $n = 1$ et $n = 2$ Si on prend $n = 1$ dans le théorème précédent on trouve le résultat suivant

Théorème 1.5.2. $\exists C > 0, \forall f \in \mathcal{C}^2([a, b]), \forall h > 0 :$

$$\max_{t \in [a, b]} |f(t) - p_1(t)| \leq Ch^2 \max_{t \in [a, b]} |f''(t)|.$$

Interprétation :

-Si $f \in \mathcal{C}^2([a, b])$: l'erreur est au moins divisée par 2^2 chaque fois que le pas h est divisé par 2.

Si on prend maintenant $n = 2$ on obtient

Théorème 1.5.3. $\exists C > 0, \forall f \in \mathcal{C}^3([a, b]), \forall h > 0 :$

$$\max_{t \in [a, b]} |f(t) - p_2(t)| \leq Ch^3 \max_{t \in [a, b]} |f'''(t)|.$$

Interprétation :

-Si $f \in \mathcal{C}^3([a, b])$: l'erreur est au moins divisée par $2^3 = 8$ chaque fois que le pas h est divisé par 2.

-Cette estimation est donc meilleure que la précédente puisque l'interpolant p_2 converge plus vite par rapport à p_1 .

On peut conclure de ce qui précède qu'il y a une dépendance entre la convergence de p_n vers f et l'ordre de régularité de f .

1.6 Interpolation d'Hermite

Dans plusieurs applications on cherche à interpoler une fonction ainsi que ses dérivées c'est à dire trouver un polynôme p tel que

$$p(t_j) = f(t_j), \quad p'(t_j) = f'(t_j) \quad \text{with } i = 0, \dots, n. \quad (1.4)$$

Dans (1.4) nous avons imposé $(2n+2)$ conditions, il est alors naturel de chercher p dans \mathbb{P}_{2n+1} .

Théorème 1.6.1. *Il existe un polynôme unique $H_{2n+1} \in \mathbb{P}_{2n+1}$ vérifiant (1.1) de plus p s'écrit sous la forme*

$$H_{2n+1}(t) = \sum_{j=0}^n f(t_j) h_j(t) + \sum_{j=0}^n f'(t_j) \tilde{h}_j(t)$$

$$\text{où } h_j(t) = [1 - 2(t - t_j)\varphi_j'(t_j)]\varphi_j^2(t) \text{ et } \tilde{h}_j(t) = (t - t_j)\varphi_j^2(t).$$

Exemple 1.6.1. Cherchons p tel que $p(a) = f(a)$, $p'(a) = f'(a)$, $p(b) = f(b)$, $p'(b) = f'(b)$. En appliquant le théorème précédent on obtient :

$$\begin{aligned} H_3(t) = & f(a) \left[1 - 2 \left(\frac{t-a}{a-b} \right) \right] \left(\frac{t-b}{a-b} \right)^2 + f(b) \left[1 - 2 \left(\frac{t-b}{b-a} \right) \right] \left(\frac{t-a}{b-a} \right)^2 \\ & + f'(a)(t-a) \left(\frac{t-b}{a-b} \right)^2 + f'(b)(t-b) \left(\frac{t-a}{a-b} \right)^2. \end{aligned}$$

Théorème 1.6.2. Si $f \in \mathcal{C}^{2n+2}([a, b])$ alors

$$f(t) - H_{2n+1} = \frac{(t-t_0)^2 \dots (t-t_n)^2}{(2n+2)!} f^{(2n+2)}(\xi)$$

avec $\xi \in [a, b]$.

Chapitre 2

Résolution numérique d'équations nonlinéaires

2.1 Position du problème

Dans ce chapitre nous allons considérer le problème de l'approximation de la solution d'une équation de type :

$$f(x) = 0$$

où f est une fonction numérique donnée

$$f : I \longrightarrow \mathbb{R}$$

L'approximation se fait généralement par des méthodes itératives où il s'agit de construire une suite (x_k) telle que :

$$\lim_{k \rightarrow \infty} x_k = \alpha$$

où α est une solution de l'équation $f(x) = 0$.

Définition 2.1.1. Soit (x_n) une suite convergente vers α . On dit que (x_n) est d'ordre p si

$$\exists C > 0 : \frac{|x_{k+1} - \alpha|}{|x_k - \alpha|^p} \leq C, \quad \forall k \geq k_0.$$

Si $p = 1$, la convergence est dite linéaire et dans ce cas $C < 1$; la constante C est la constante asymptotique de convergence ou de l'erreur.

2.2 Méthode de dichotomie ou de bisection

Cette méthode est une application directe du théorème des valeurs intermédiaires.

Théorème 2.2.1 (Théorème des valeurs intermédiaires). Soit f une fonction continue sur $I = [a, b]$ tels que

$$\inf_I f(x) = m \quad \text{et} \quad \sup_I f(x) = M$$

alors : $\forall \xi \in [m, M] \exists c \in I$ tels que $f(c) = \xi$.

On en déduit du théorème précédent que si $f(a).f(b) < 0$ alors $\exists \alpha \in [a, b]$ tel que $f(\alpha) = 0$.

Supposons que $[a, b]$ contient une seule racine α ; alors on procède de la manière suivante :

on pose $a_1 = a$; $b_1 = b$ et $c_1 = \frac{a_1+b_1}{2}$

si $f(a_1).f(c_1) > 0$ alors $\alpha \in [c_1, b_1]$

on pose $a_2 = c_1$; $b_2 = b_1$ et on applique le même procédé pour $[a_2, b_2]$

sinon $\alpha \in [a_1, c_1] \rightarrow a_2 = a_1; b_2 = c_1$ etc...

Ainsi à chaque étape l'intervalle contenant α est réduit d'un facteur de 2. D'où l'algorithme :

Données : f, a, b, ϵ

Sortie : $\tilde{\alpha}$ (approximation du zéro de f sur $[a, b]$).

Etape 1 : $c = \frac{a+b}{2}$ (génération de la suite (c_n)).

Etape 2 : Si $|b - c| < \epsilon$ alors $\tilde{\alpha} := c$. Stop.

Etape 3 : Si $f(a).f(c) \leq 0$ alors $a := c$

Sinon $b := c$

Etape 4 : Aller à l'étape 1.

Théorème 2.2.2 (Convergence de la méthode de dichotomie). Soit (c_n) la suite générée par l'algorithme précédent, alors :

$$\lim_n c_n = \alpha \quad \text{et} \quad |\alpha - c_n| \leq \frac{b-a}{2^n}.$$

Preuve. Nous avons :

$$|b_2 - a_2| = \frac{b_1 - a_1}{2}$$

et

$$|b_n - a_n| = \frac{b_{n-1} - a_{n-1}}{2} = \dots = |b_2 - a_2| = \frac{b-a}{2^{n-1}}$$

d'où

$$|\alpha - c_n| \leq \frac{b_n - a_n}{2} = \frac{b-a}{2^n}$$

□

Tests d'arrêts : Différents tests sont possible, par exemple :

- $|c_n - c_{n-1}| < \epsilon$

- $\frac{|c_n - c_{n-1}|}{c_n}$ si $c_n \neq 0$ qui est meilleur que le précédent.

- $f(c_n) < \epsilon$

Remarque 2.2.1. La méthode de bisection converge très lentement en général, mais elle a l'avantage d'être toujours convergente dans les conditions citées ci-dessus. Elle est alors utilisée pour initialiser des méthodes plus rapides.

2.3 Méthode de Newton-Raphston

Soit à chercher α racine de $f(x) = 0$ et soit x_0 choisit dans un voisinage de α .

La méthode de Newton consiste à construire à partir de x_0 une suite (x_n) convergente vers α , en remplaçant le graphe de $y = f(x)$ au voisinage de α par la tangente au point $(x_n, f(x_n))$. L'équation de la tangente est donnée par :

$$y - f(x_n) = f'(x_n)(x - x_n) \quad \text{d'où l'itération}$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad n \geq 0 \quad (2.1)$$

C'est la méthode la plus connue pour approcher la solution de $f(x) = 0$. On peut retrouver (2.1) en utilisant la formule de Taylor de la manière suivante : au voisinage de x_n on a :

$$f(x) = f(x_n) + f'(x_n) \cdot (x - x_n) + f''(\xi_n) \cdot \frac{(x - x_n)^2}{2}$$

pour $x = \alpha$ et comme $f(\alpha) = 0$ on en déduit que :

$$\alpha = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f''(\xi_n)}{f'(x_n)} \cdot \frac{(\alpha - x_n)^2}{2}$$

en négligeant le terme d'erreur, nous obtenons une meilleure approximation de α que x_n ce qui permet de retrouver l'itération (2.1).

Théorème 2.3.1 (Convergence de la méthode de Newton). *Supposons que la fonction f est de classe \mathcal{C}^2 au voisinage de α et que $f(\alpha) = 0$; $f'(\alpha) \neq 0$. Alors $\exists \epsilon > 0$ tel que si $x_0 \in [\alpha - \epsilon, \alpha + \epsilon]$ la suite (x_n) définie par (2.1) est bien définie et converge vers α . De plus on a :*

$$\lim_{n \rightarrow \infty} \frac{(\alpha - x_{n+1})}{(\alpha - x_n)^2} = -\frac{f''(\alpha)}{2f'(\alpha)} \quad (2.2)$$

ce qui montre que la méthode est d'ordre 2.

Preuve. Comme f est continue $\exists \epsilon$ tel que $f'(x) \neq 0$ sur $I = [\alpha - \epsilon, \alpha + \epsilon]$, soit

$$M = \frac{\max_I |f''(x)|}{2 \min_I |f'(x)|}$$

on a

$$|\alpha - x_1| = (\alpha - x_0)^2 \frac{|f''(\xi)|}{2|f'(x_0)|}$$

$$\leq (\alpha - x_0)^2 M$$

d'où si $\epsilon < \frac{1}{M}$ alors $|\alpha - x_1| < \epsilon$ et par récurrence on obtient que $\forall n \quad |\alpha - x_n| < \epsilon$; $M|\alpha - x_n| < 1$; donc la suite (x_n) est bien définie.

D'autre part :

$$M|\alpha - x_{n+1}| \leq (M|\alpha - x_n|)^2$$

$$\Rightarrow M|\alpha - x_n| \leq (M|\alpha - x_0|)^{2^n}$$

et $M|\alpha - x_0| < 1$ d'où $x_n \rightarrow \alpha$ et de plus nous avons d'après la formule de Taylor à l'ordre 2 :

$$\frac{(\alpha - x_{n+1})}{(\alpha - x_n)^2} = -\frac{f''(\xi_n)}{2f'(x_n)} \quad \text{avec} \quad x_n < \xi_n < x_{n+1}$$

d'où (2.2). □

Tests d'arrêts : On peut utiliser les mêmes tests que pour la bisection, mais pour Newton on peut remarquer que :

$$f(x_n) = f(x_n) - f(\alpha) = f'(\xi_n)(x_n - \alpha) \Rightarrow x_n - \alpha = \frac{f(x_n)}{f'(\xi_n)}$$

si f' ne varie pas rapidement au voisinage de α , on peut supposer que $f'(\xi_n) \simeq f'(x_n)$, d'où

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

d'où le test standard pour la méthode de Newton

$$|\alpha - x_n| \simeq |x_{n+1} - x_n| < \epsilon$$

et le test relatif correspondant

$$\frac{|x_{n+1} - x_n|}{|x_{n+1}|} < \epsilon$$

Algorithme de Newton :

Données $(f, fpm, x_0, itmax, iter)$

1. $iter = 1$
2. $fpm = f'(x_0)$
3. si $fpm = 0$, $iter = 2$ et sortir
4. $x1 := x_0 - \frac{f(x_0)}{fpm}$
5. si $|x1 - x_0| < \epsilon$ alors $iter = 0$; *racine* = $x1$. Stop
6. si $iter = itmax$: 'la méthode est divergente', $iter := 1$
7. $iter := iter + 1$; $x_0 = x1$ aller à 2.

Remarque 2.3.1. En général quand la méthode de Newton converge, elle converge très rapidement mais souvent le choix de x_0 n'est pas facile à faire, on a besoin des deux fonctions f et f' ce qui ralentit aussi parfois la méthode.

2.4 Théorème du point fixe et méthode d'approximations successives

Dans cette section l'équation $f(x) = 0$ est considérée sous la forme équivalente

$$g(x) - x = 0$$

où $g : [a, b] \longrightarrow \mathbb{R}$ une fonction auxiliaire telle que $g(\alpha) = \alpha$.

Une méthode très simple pour approcher les solutions de $g(x) = x$ et l'itération du point fixe suivante : Soit x_0 un point initial donné et la suite (x_n) définit par :

$$x_{n+1} = g(x_n) \quad n \geq 0$$

Exemple 2.4.1. La racine carée d'un réel $a > 0$ est solution de $f(x) = x^2 - a = 0$. Pour approcher \sqrt{a} on peut considérer l'une des fonctions g :

(a) $g(x) = x^2 + x - a$

(b) $g(x) = \frac{a}{x}$

(c) $g(x) = \frac{1}{2}(x + \frac{a}{x})$

Remarque 2.4.1. Le choix de la fonction g n'est pas unique.

Lemme 2.4.1. Soit g une fonction continue de l'intervalle $[a, b]$ dans lui même ($g([a, b]) \subset [a, b]$), alors g admet au moins un point fixe dans $[a, b]$.

Preuve. La fonction $h(x) = g(x) - x$ vérifie $h(a).h(b) \leq 0$ et elle est continue sur $[a, b]$, donc on peut déduire, d'après le théorème des valeurs intermédiaires, qu'il existe au moins un point $x_0 \in [a, b]$ tel que $h(x_0) = 0$ soit $g(x_0) = x_0$ d'où le résultat. \square

Théorème 2.4.1. Soit g une fonction continue de l'intervalle $[a, b]$ dans lui même vérifiant

$$\exists 0 \leq \lambda < 1 \quad \text{telle que } |g(x) - g(y)| \leq \lambda|x - y| \quad \forall (x, y) \in [a, b]^2, \quad (2.3)$$

alors $g(\cdot)$ admet un point fixe α **unique** dans $[a, b]$, l'itération

$$x_{n+1} = g(x_n) \quad n \geq 0$$

converge vers α pour tout $x_0 \in [a, b]$ et on a l'estimation :

$$|\alpha - x_n| \leq \frac{\lambda^n}{1 - \lambda} |x_1 - x_0|. \quad (2.4)$$

Preuve. -Unicité : Supposons qu'il existe deux points fixes $\alpha \neq \beta$. D'après l'hypothèse (2.3) on a

$$\begin{aligned} |g(\alpha) - g(\beta)| &\leq \lambda|\alpha - \beta| \implies |\alpha - \beta| \leq \lambda|\alpha - \beta| \\ &\implies \lambda \geq 1 \end{aligned}$$

ce qui contredit l'hypothèse $\lambda < 1$ d'où le résultat d'unicité.

D'autre part on a :

$$\begin{aligned} |\alpha - x_{n+1}| &= |g(\alpha) - g(x_n)| \implies |\alpha - x_{n+1}| \leq \lambda|\alpha - x_n| \\ &\implies |\alpha - x_n| \leq \lambda^n |\alpha - x_0| \end{aligned}$$

et comme $0 \leq \lambda < 1$ on a bien $\lim_{n \rightarrow \infty} |\alpha - x_n| = 0$ d'où la convergence.

En utilisant l'inégalité triangulaire on obtient :

$$|\alpha - x_0| \leq |\alpha - x_1| + |x_1 - x_0| \leq \lambda|\alpha - x_0| + |x_1 - x_0|$$

d'où : $|\alpha - x_0| \leq \frac{|x_1 - x_0|}{1 - \lambda}$ et en reportant dans l'inégalité ci-dessus on obtient (2.4). \square

Nous avons aussi le théorème similaire suivant :

Théorème 2.4.2. Soit g une fonction de classe \mathcal{C}^1 de l'intervalle $[a, b]$ dans lui même et

$$\lambda = \max_{a \leq x \leq b} |g'(x)| < 1.$$

alors nous avons les mêmes résultats que le théorème précédent avec en plus :

$$\lim_{n \rightarrow \infty} \frac{|\alpha - x_{n+1}|}{|\alpha - x_n|} = g'(\alpha). \quad (2.5)$$

C'est à dire que la convergence est linéaire.

Preuve. En utilisant le théorème des accroissements finis on montre que

$$|g(x) - g(y)| \leq \lambda |x - y| \quad \forall (x, y) \in [a, b]^2$$

et il suffit alors d'appliquer le théorème précédent, de plus on a :

$$\frac{\alpha - x_{n+1}}{\alpha - x_n} = g'(\xi_n) \text{ avec } \xi_n \in [\alpha, x_n]$$

comme $g'(\cdot)$ est continue sur $[a, b]$ on peut passer à la limite sur n et obtenir (2.5). \square

On peut démontrer aussi le théorème suivant :

Théorème 2.4.3. Soit g une fonction de classe \mathcal{C}^1 au voisinage d'un point fixe α avec $g'(\alpha) < 1$. Alors il existe un voisinage V_α de α tel que si $x_0 \in V_\alpha$, l'itération du point fixe $x_{n+1} = g(x_n)$, $n \geq 0$ est convergente et (2.5) est vérifiée.

Algorithme de la méthode du point fixe :

-Données $(g, x_0, \epsilon, itmax)$

-Sortie α le point fixe de g

1. $iter := 1; err = 1 + \epsilon$
2. Tant que $(iter \leq itmax \text{ et } err > \epsilon)$ faire
 - (a) $xk = g(x_0)$
 - (b) $err := |xk - x_0|$
 - (c) $iter = iter + 1$
 - (d) $x_0 = xk$
3. Si $(iter > itmax)$ alors la méthode est divergente.
Sinon $\alpha = xk$.

Chapitre 3

Résolution d'un système linéaire

3.1 Position du problème

Dans ce chapitre, nous considérons un système d'équations linéaire d'ordre n de la forme

$$A\vec{x} = \vec{b}. \quad (3.1)$$

Ici A est une matrice $n \times n$ régulière (invertible) d'ordre n de coefficients a_{ij} , $1 \leq i, j \leq n$, donnés, \vec{b} est un vecteur colonne à n composantes b_j , $1 \leq j \leq n$, données et \vec{x} est un vecteur colonne à n composantes x_j , $1 \leq j \leq n$, inconnues. Dans la suite nous utilisons les notations matricielles standards, i.e.,

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

Définition 3.1.1. On dira que la matrice A est triangulaire supérieure (respectivement triangulaire inférieure) si $a_{ij} = 0$ pour tout i, j tel que $1 \leq j < i \leq n$ (resp. $1 \leq i < j \leq n$).

Supposons que la matrice A soit triangulaire supérieure régulière. La résolution du système linéaire $A\vec{x} = \vec{b}$:

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

se fait par *l'algorithme* très simple dit *de remontée* suivant :

$$x_n = \frac{b_n}{a_{nn}}$$

$$x_i = \frac{1}{a_{ii}} \left(b_i - \sum_{k=n}^{i+1} a_{ik} x_k \right), \quad i = n-1, \dots, 1$$

cet algorithme nécessite le nombre d'opérations élémentaires :

$$\sum_{i=1}^{n-1} (n-i) = 1 + 2 + \dots + n-1 = \frac{n(n-1)}{2} \quad \text{additions}$$

$$\sum_{i=1}^n (n-i) = 1 + 2 + \dots + n-1 = \frac{n(n-1)}{2} \quad \text{multiplications}$$

$$n \quad \text{divisions}$$

Exercice 3.1.1. *Ecrire l'algorithme correspondant à un système triangulaire inférieure (S.T.I) et celui correspondant à un système triangulaire supérieure (S.T.S).*

La *méthode de Gauss* est une **méthode directe** de résolution d'un système linéaire. Une méthode directe conduit à la solution exacte (sans les erreurs d'arrondis) en un nombre fini d'opérations élémentaires. Le principe de la méthode de Gauss revient à la détermination d'une matrice inversible M telle que la matrice MA soit triangulaire supérieure, la résolution du système $Ax = b$ se réduit à celle de $MAx = Mb$ qui est facile à résoudre.

3.2 Méthode de Gauss sur un exemple

Considérons un exemple avec $n = 3$ pour montrer comment faire la transformation d'un système linéaire en un système triangulaire supérieur équivalent.

$$\begin{array}{ll} L_1 & 2x_1 + x_2 + x_3 = 4 \\ L_2 & x_1 + 2x_2 + x_3 = 4 \\ L_3 & 3x_1 + x_2 + 4x_3 = 8 \end{array}$$

$$A = A_1 = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 3 & 1 & 4 \end{pmatrix}, \quad b_1 = \begin{pmatrix} 4 \\ 4 \\ 8 \end{pmatrix}.$$

Etape 1 :

$$\begin{array}{ll} L_1^{(1)} & 2x_1 + x_2 + x_3 = 4 \\ L_2 - \frac{L_1}{2} & 0x_1 + \frac{3}{2}x_2 + \frac{1}{2}x_3 = 2 \\ L_3 - \frac{3}{2}L_1 & 0x_1 - \frac{1}{2}x_2 + \frac{5}{2}x_3 = 2 \end{array}$$

$$A_2 = \begin{pmatrix} 2 & 1 & 1 \\ 0 & \frac{3}{2} & \frac{1}{2} \\ 0 & -\frac{1}{2} & \frac{5}{2} \end{pmatrix}, \quad b_2 = \begin{pmatrix} 4 \\ 2 \\ 2 \end{pmatrix}.$$

Remarque 3.2.1. Lors de cette étape nous avons effectué le produit matriciel suivant

$$\begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ -\frac{3}{2} & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 3 & 1 & 4 \end{pmatrix} \text{ et } \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ -\frac{3}{2} & 0 & 1 \end{pmatrix} \begin{pmatrix} 4 \\ 4 \\ 8 \end{pmatrix}.$$

Etape 2 :

$$\begin{aligned} L_1^{(2)} & & 2x_1 + x_2 + x_3 &= 4 \\ L_2^{(2)} - \frac{L_1}{2} & & 0x_1 + \frac{3}{2}x_2 + \frac{1}{2}x_3 &= 2 \\ L_3^{(2)} - \frac{2}{3} \times \frac{-1}{2}L_2^{(2)} & & 0x_1 + 0x_2 + \frac{8}{3}x_3 &= \frac{8}{3} \end{aligned}$$

$$A_3 = \begin{pmatrix} 2 & 1 & 1 \\ 0 & \frac{3}{2} & \frac{1}{2} \\ 0 & 0 & \frac{8}{3} \end{pmatrix}, \quad b_3 = \begin{pmatrix} 4 \\ 2 \\ \frac{8}{3} \end{pmatrix}.$$

Remarque 3.2.2. Lors de cette étape nous avons effectué le produit matriciel suivant

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{3} & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 1 \\ 0 & \frac{3}{2} & \frac{1}{2} \\ 0 & -\frac{1}{2} & \frac{5}{2} \end{pmatrix} \text{ et } \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{3} & 1 \end{pmatrix} \begin{pmatrix} 4 \\ 2 \\ 2 \end{pmatrix}.$$

Remarque 3.2.3. Nous n'avons pas calculé explicitement la matrice M mais MA et Mb .

Exercice 3.2.1. Refaites la méthode de Gauss pour le système suivant $Ax = b$ avec $A = \begin{pmatrix} 4 & 8 & 12 \\ 3 & 8 & 13 \\ 2 & 9 & 18 \end{pmatrix}$ et $b = \begin{pmatrix} 4 \\ 5 \\ 11 \end{pmatrix}$.

3.2.1 Algorithme d'élimination de Gauss

Etape 1 : Sur la première colonne de A il existe au moins un élément non nul a_{i1} sinon A n'est pas régulière. On choisit alors ce coefficient que nous appellerons le premier pivot de l'élimination et on échange la ligne du pivot avec la première ligne.

Nous faisons apparaître des zéros sous a_{11} en combinant la première et la i ème ligne par

$$\begin{aligned} a_{ij}^{(2)} &= a_{ij}^1 - \frac{a_{i1}^1 a_{1j}^1}{a_{11}^1} \quad \text{pour } j = 1, \dots, n; i = 2, \dots, n \\ b_i^{(2)} &= b_i^1 - \frac{a_{i1}^1 b_1^{(1)}}{a_{11}^1} \end{aligned}$$

cela revient à multiplier A_1 par la matrice

$$L_1 = \begin{pmatrix} 1 & 0 & \dots & 0 \\ -\frac{a_{21}^1}{a_{11}^1} & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ -\frac{a_{n1}^1}{a_{11}^1} & 0 & \dots & 1 \end{pmatrix}.$$

Nous obtenons alors

$$A_2 = L_1 P A_1 = \begin{pmatrix} a_{11}^1 & \cdots & \cdots & a_{1n}^1 \\ 0 & a_{22}^2 & \cdots & a_{2n}^2 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & a_{n2}^2 & \cdots & a_{nn}^2 \end{pmatrix}.$$

On a le $\det L_1 = 1$ d'où $\det A_2 = \pm \det A$ donc A_2 est aussi inversible et on peut alors refaire la même opération sur A_2 . Supposons que de proche en proche on arrive à la (k-1)ème étape de l'élimination à la matrice :

$$A_k = L_{k-1} P_{k-1} \cdots L_2 P_2 L_1 P_1 A_1$$

de la forme

$$\begin{pmatrix} a_{11}^k & \cdots & \cdots & a_{1n}^k \\ \vdots & \ddots & \ddots & \vdots \\ \cdots & a_{kk}^k & \cdots & a_{kn}^k \\ \vdots & \vdots & \cdots & \vdots \\ \cdots & a_{nk}^k & \cdots & a_{nn}^k \end{pmatrix}$$

Etape k : Echange de la k-ième ligne avec la ligne du pivot ($a_{ik}^k \neq 0$) soit matriciellement :

$$A'_k = P_k A_k.$$

On notera dans la suite de la même façon A'_k et A_k .-k-ième étape de l'élimination

$$a_{ij}^{(k+1)} = a_{ij}^k - \frac{a_{ik}^k a_{kj}^k}{a_{kk}^k} \text{ pour } j = k+1, \dots, n; i = k+1, \dots, n$$

$$b_i^{(k+1)} = b_i^{(k)} - \frac{a_{ik}^k b_i^{(k)}}{a_{kk}^k}$$

ce qui revient à multiplier à gauche par la matrice

$$L_k = \begin{pmatrix} 1 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \cdots & 1 & \cdots & 0 \\ \vdots & -l_{k+1,k} & 1 & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ \cdots & -l_{nk} & \cdots & 1 \end{pmatrix}.$$

Après (n-1) étape nous obtenons la matrice triangulaire supérieure

$$A_n = L_{n-1} P_{n-1} \cdots L_2 P_2 L_1 P_1 A$$

d'où la matrice

$$M = L_{n-1} P_{n-1} \cdots L_2 P_2 L_1 P_1.$$

On a suivant la parité du nombre d'échange de ligne effectuée $\det M = \pm 1$, ce qui permet d'obtenir le déterminant de A par :

$$\det A = \frac{\det A_n}{\det M} = \pm a_{11}^{r_1} a_{22}^{r_2} \cdots a_{nn}^{r_n}.$$

On peut résumer ce qui précède par le théorème suivant

Théorème 3.2.1. *Soit A une matrice carrée. Il existe une matrice inversible M telle que la matrice MA soit triangulaire supérieure.*

Preuve. Nous venons de montrer le résultat du théorème pour A inversible. La matrice A ne serait pas inversible s'il existe k tel que au cours de la k -ième étape tous les a_{ik}^k pour $k \leq i \leq n$ de A_k sont nuls ; il suffit alors de passer à l'étape $k+1$ en posant $P_k = L_k = I$ soit $A_{k+1} = A_k$. \square

Algorithme d'élimination de Gauss sans pivotage :

-**Données :** Dimension du système n , matrice A augmentée du second membre $b = A[i, n+1]$, $i = 1, \dots, n$

-**Résultats :** Solutions x_1, \dots, x_n ou message : 'pas de solutions unique'.

-**étape 1 :** Pour $k = 1, \dots, n-1$ faire

-**étape 2 :** Soit p le plus petit entier tel que $0 \leq p \leq n$ et $A[p, k] \neq 0$.

Si p n'existe pas ; sortir : 'pas de solutions unique'

Stop.

-**étape 3 :** Si $p \neq k$ faire l'échange $L_p \longleftrightarrow L_k$

-**étape 4 :** Pour $i = k+1, \dots, n$ faire

-**étape 5 :** $\alpha = \frac{A[i, k]}{A[k, k]}$

-**étape 6 :** Pour $j = k+1, \dots, n+1$ faire

$A[i, j] = A[i, j] - \alpha * A[k, j]$

-**étape 7 :** Si $A[n, n] = 0$ sortir : 'pas de solutions unique'

Stop.

-**étape 8 :** $x[n] = \frac{A[n, n+1]}{A[n, n]}$

-**étape 9 :** Pour $i = n-1, \dots, 1$ faire

$x[i] = \frac{A[i, n+1] - \sum_{j=i+1}^n A[i, j]x[j]}{A[i, i]}$

-**étape 10 :** Sortir $x[1], \dots, x[n]$

Stop.

3.2.2 Compte des opérations élémentaires dans la méthode de Gauss

Ce sont les opérations nécessaires pour le passage de A_k à A_{k+1} ; $1 \leq k \leq n+1$ plus la remontée du système. L'algorithme étant (sans tenir compte des échanges de lignes) :

$$a_{ij}^{k+1} = a_{ij}^k - \frac{a_{ik}^k a_{kj}^k}{a_{kk}^k} \text{ pour } j = k+1, \dots, n; i = k+1, \dots, n$$

$$b_i^{k+1} = b_i^{(k)} - \frac{a_{ik}^k b_k^{(k)}}{a_{kk}^k} \text{ pour } k = 1, \dots, n.$$

Le passage de A_k à A_{k+1} nécessite $(n-k)(n-k+1)$ multiplications, autant d'additions et $(n-k)$ divisions ; d'où le nombre total d'opérations lors de l'élimination :

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

additions et multiplications et

$$\sum_{k=1}^{n-1} n - k = \frac{n(n-1)}{2} \quad \text{divisions.}$$

Le nombre d'opérations pour la remontée étant $n(n-1) + n$, la méthode de Gauss nécessite donc de l'ordre de $\frac{2}{3}n^3$ opérations élémentaires.

-Comparaison avec les formules de Cramer : on a $x_j = \frac{\det B_i}{\det A}$, B_i est la matrice A où on remplace la i -ème colonne par le vecteur b .

$$\det A = \sum_{\sigma} \varepsilon(\sigma) \prod_{i=1}^n a_{i\sigma(i)}$$

le nombre de permutations possible étant $n!$ le calcul du déterminant nécessite $n!(n-1) + n! - 1 = nn! - 1$, donc pour calculer toutes les composantes il faut $n(n+1)!$ opérations. Par exemple pour $n = 100$ sur un superordinateur (10^8 opérations par seconde) il faudrait $9,4 * 10^{161}$ opérations donc il faudrait à peu près $3 * 10^{146}$ années (10^{135} fois l'âge de l'univers).

Avec la méthode de Gauss : $\frac{2}{3}10^6$ opérations soit $\frac{2}{3}10^{-2}$ seconde.

3.3 Décomposition LU d'une matrice

Supposons que dans la démonstration du théorème précédent on n'a pas eu à faire d'échange de ligne ; les $P_i = I$; alors on aurait :

$$A_n = L_{n-1} \cdots L_2 L_1 A$$

où A_n est une matrice triangulaire supérieure (Upper) ; les matrices L_i sont régulières donc on peut poser

$$L = L_1^{-1} L_2^{-1} \cdots L_{n-1}^{-1}$$

et alors $A = LU$ avec L est une matrice triangulaire inférieure (Lower) tel que $l_{ii} = 1$ pour $1 \leq i \leq n$. On peut expliciter la matrice L , en effet :

$$L_k^{-1} = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & l_{k+1,k} & 1 & \\ & & \vdots & & \ddots \\ & & l_{nk} & & & 1 \end{pmatrix}$$

$$\Rightarrow L = L_1^{-1} L_2^{-1} \cdots L_{n-1}^{-1} = \begin{pmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \vdots & \vdots & \ddots & 1 & \\ \vdots & \vdots & & \ddots & \\ l_{n1} & l_{n2} & \cdots & \cdots & l_{n,n-1} & 1 \end{pmatrix}.$$

Remarque 3.3.1. On peut pas expliciter aussi simplement la matrice M dans ce cas.

Théorème 3.3.1. Soit A une matrice carrée régulière d'ordre n . Les deux assertions suivantes sont équivalentes :

- i) Il existe une matrice triangulaire inférieure $L = (l_{ij})$ avec $l_{ii} = 1$, et une unique matrice triangulaire supérieure U telles que

$$A = LU$$

- ii) Tous les mineurs principaux de A , $\det A_k = \begin{vmatrix} a_{11} & \cdots & a_{1k} \\ \vdots & \ddots & \vdots \\ a_{k1} & \cdots & a_{kk} \end{vmatrix}$, sont non nuls.

Si l'un des mineurs principaux est nul et A est régulière, on a le théorème suivant :

Théorème 3.3.2. Soit A une matrice régulière. Il existe une matrice de permutation P telle que la matrice PA admet une factorisation LU .

Application : Si on possède une décomposition LU de A , pour résoudre $Ax = b$: on résout les deux systèmes triangulaires $Ly = b$ puis $Ux = y$. Ceci est intéressant quand on a plusieurs systèmes à résoudre avec la même matrice.

Exercice 3.3.1. Donner la décomposition LU de la matrice A et résoudre le système $Ax = b$ en utilisant les algorithmes STS et STI.

$$A = \begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 11 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

Algorithme de factorisation LU :

Entrées : La dimension n , les composantes de $A = (a_{ij})_{1 \leq i, j \leq n}$.

Sorties : Les composantes l_{ij} de L ; $1 \leq j \leq i$, $1 \leq i \leq n$ et celles de $U = (u_{ij})_{1 \leq i, j \leq n}$.

Etape 1 : $u_{11} = a_{11}$.

Si $u_{11} = 0$ alors afficher : 'La factorization est impossible'

Stop.

Etape 2 : Pour $j = 2, \dots, n$

$u_{1j} = a_{1j}$ (Première ligne de U) $l_{j1} := \frac{a_{j1}}{u_{11}}$ (Première colonne de L)

Etape 3 : Pour $i = 2, \dots, n-1$ faire les étapes 4-5.

Etape 4 : $u_{ii} = a_{ii} - \sum_{k=1}^{i-1} l_{ik} u_{ki}$.

Si $u_{ii} = 0$ alors afficher : 'La factorization est impossible'

Stop.

Etape 5 : Pour $j = i+1, \dots, n$

$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj}$ (La i^{eme} ligne de U .)

$l_{ji} = \frac{1}{u_{ii}} \left[a_{ji} - \sum_{k=1}^{i-1} l_{jk} u_{ki} \right]$ (La i^{eme} colonne de L .)

Etape 6 : $u_{nn} = a_{nn} - \sum_{k=1}^{n-1} l_{nk} u_{kn}$ (Rq : Si $u_{nn} = 0$ alors $A = LU$ mais A non inversible.)

Etape 7 : Sortir (l_{ij} Pour $j = 1, \dots, i$ et pour $i = 1, \dots, n$);

Sortir (u_{ij} Pour $j = i \cdots, n$ et pour $i = 1, \cdots, n$)

Stop.

En pratique les composantes u_{ij} et l_{ij} sont stockés à la place des a_{ij} ce qui donne lieu à l'**algorithme de décomposition optimisant la place mémoire** :

Fonction *decompositionLU*(A, n)

Pour $k = 1, n - 1$

Pour $i = k + 1, n$

$facteur = \frac{a_{ik}}{a_{kk}}$

$a_{ik} = facteur$

Pour $j = k + 1, n$

$a_{ij} = a_{ij} - facteur * a_{kj}$

Fin pour j

Fin pour i

Fin pour k

Fin *decomposition*

3.4 Décomposition de Cholesky

Définition 3.4.1. *Une matrice A d'ordre n est dite symétrique définie positive si*

- i) $A = A^T$ (A est symétrique),
- ii) $y^T A y \geq 0$ pour tout vecteur $y \in \mathbb{R}^n$.
- ii) $y^T A y = 0$ si et seulement si $y = 0$.

Théorème 3.4.1. *Si A est une matrice d'ordre n symétrique définie positive, alors toutes ses sous matrices principales sont symétriques définies positives et sont donc régulières.*

Théorème 3.4.2 (Théorème de Cholesky). *Si A est une matrice symétrique définie positive, il existe une et une seule matrice triangulaire inférieure à valeurs diagonales positives notée L telle que $A = LL^T$.*

Remarque 3.4.1. *Dans le théorème précédent nous avons noté L la matrice triangulaire inférieure qui satisfait $A = LL^T$. Evidemment, cette matrice L n'est en principe pas la même que celle obtenue lors de la décomposition LU.*

En conclusion, lorsque on veut résoudre un système linéaire $Ax = b$, où la matrice A est symétrique définie positive, on fait une décomposition de Cholesky $A = LL^T$ et on résout successivement

$$Ly = b, \quad \text{puis } L^T x = y.$$

L'algorithme de Cholesky consiste donc à calculer explicitement, la matrice triangulaire inférieure L . Ce calcul peut se faire par identification des coefficients situés dans la partie triangulaire inférieure de la matrice A .

Comme

$$a_{ij} = \sum_{k=1}^j b_{ik} b_{jk}, \quad j \geq i$$

et compte tenu de la forme triangulaire de L . On peut calculer les éléments de la première colonne de L :

$$\begin{aligned} a_{11} &= b_{11}^2 \Rightarrow b_{11} = \sqrt{a_{11}}, \\ a_{i1} &= b_{i1}b_{11} \Rightarrow b_{i1} = \frac{a_{i1}}{b_{11}} \end{aligned}$$

Supposons calculées les j premières colonnes de L , on calcule sa j^{ieme} colonne en écrivant

$$\begin{aligned} b_{jj} &= \left(a_{jj} - \sum_{k=1}^{j-1} b_{jk}^2 \right)^{1/2} \\ b_{ij} &= \left(\frac{a_{ij} - \sum_{k=1}^{j-1} b_{ik}b_{jk}}{b_{jj}} \right), \quad i \geq j+1 \end{aligned}$$

Remarque 3.4.2. 1. Si A n'est pas définie positive, alors il existe au moins un b_{kk}^2 négatif. Cette méthode permet de savoir si une matrice symétrique est définie positive.

2. Cette méthode permet en même temps de calculer le déterminant de A :

$$\det A = \det(LL^T) = \prod_{i=1}^n b_{ii}^2.$$

Nombre d'opérations élémentaires :

La méthode de Cholesky nécessite :

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

$\sum_{i=1}^n (n-i) = \frac{n^2-n}{2}$ divisions,

n racines carrées.

Algorithme de Cholesky :

Pour factoriser une matrice A sous forme $A = LL^T$

Entrées : n , la dimension de A , les composantes $(a_{ij})_{1 \leq i, j \leq n}$; de la matrice A .

Sorties : Les composantes (l_{ij}) , $1 \leq j \leq i$, $1 \leq i \leq n$ de L .

Etape 1 : $b_{11} = \sqrt{a_{11}}$.

Etape 2 : Pour $i = 2, \dots, n$; $b_{i1} = \frac{a_{i1}}{b_{11}}$.

Etape 3 : Pour $j = 2, \dots, n-1$ faire les étapes 4-5.

Etape 4 : $b_{jj} = \left(a_{jj} - \sum_{k=1}^{j-1} b_{jk}^2 \right)^{1/2}$.

Etape 5 : Pour $i = j+1, \dots, n$

$$b_{ij} = \left(\frac{a_{ij} - \sum_{k=1}^{j-1} b_{ik}b_{jk}}{b_{jj}} \right)$$

Etape 6 : $b_{nn} = \left(a_{nn} - \sum_{k=1}^{n-1} b_{nk}^2 \right)^{1/2}$.

Etape 7 : Sortir (b_{ij}) , pour $i = 1, \dots, j$ et $j = 1, \dots, n$;

Stop.