Ministère de l'Enseignement Supérieur, de la Recherche Scientifique et de la Technologie
Ecole Supérieure de la Statistique et de l'Analyse de l'Information de Tunisie

Cours de Calcul Scientifique

Analyse Numérique

Version Provisoire

Première Année du cycle d'Ingénieur

Ines Abdeljaoued-TEJ

Année Universitaire 2007-2008

ESSAI, située à l'INSAT, Zone urbaine nord - BP 675, Tunis 1080
Tél : 71 703 717 / 71 704 309 - Poste 6220 - **Fax :** 71 704 329
Mél : Ines.Abdeljaoued@insat.rnu.tn

Objectifs Il s'agit de décrire des outils et des méthodes constamment utilisées en Analyse Numérique. La maîtrise de ces méthodes est une nécessité pour la résolution des problèmes que l'ingénieur est amené à résoudre au cours de sa vie professionnelle. L'étude des algorithmes permettra de comprendre la notion de stabilité, de précision, de complexité et de choisir la méthode assurant le meilleur coût en terme de temps de mémoire nécessaires pour la résolution de problèmes numériques.

Pré-requis Notions élémentaires d'algorithmique, calculs matriciels (determinant, valeurs propres, ...), résolutions de système d'équations (élimination de Gauss), étude de fonctions, etc...

Programme Le volume horaire de ce module est de 18h de Cours, 10h de TD et 6h de TP. La présence aux séances de Cours, de TD et de TP est obligatoire. Le plan d'enseignement est le suivant :

1. Interpolation et Approximation
   (a) Formule d'interpolation de Lagrange
   (b) La méthode des différences divisées
   (c) Estimation de l'erreur
   (d) Interpolation Spline
   (e) Introduction à l'approximation
2. Equations non-linéaires à une variable
   (a) Méthode de bissection
   (b) Méthode de Newton
   (c) Méthode de la sécante
   (d) Méthode du point fixe
   (e) Accélération de la convergence
3. Analyse numérique matricielle
   (a) Une méthode directe de résolution de système : Gauss
   (b) La factorisation LU d'une matrice
   (c) La factorisation QR
   (d) Une méthode itérative : Jacobi
   (e) Calcul des valeurs et vecteurs propres

Contrôle des connaissances Trois interrogations de 30 minutes (1 pour chacune des 3 parties du Cours) et1 test sur machine (lors de la dernière séance de TP). La moyenne des notes définie la note de Contrôle Continu.
Un examen final de 2h00 qui compte pour 65% de la note finale.

Lors de la session de rattrapage, la note de CC est conservée et la note de l'examen est remplacée par le sup de la note de l'examen de contrôle et l'examen de la session principale.

Bibliographie Cette liste n'est pas exhaustive. Vous trouverez à la bibliothèque des livres de programmation en Matlab, Scilab, Maple et Mupad : ce sont des logiciels de calcul numérique et de calcul formel.

B. Demidovitch et I. Maron,*Eléments de Calculs Numériques, MIR, MOSCOU.*

S. Mathlouthi, Analyse numérique linéaire et non linéaire, Collections M / Sciences fondamentales, CPU.

J. Bastien et J-N. Matin, Introduction à l'analyse numérique - Applications sous Matlab, Editions Dunod, Paris.

P.G. Ciarlet, Introduction à l'analyse numérique matricielle et à l'optimisation, Masson.

P.G. Ciarlet et B. Miara et J-M. Thomas, Exercices d'analyse numérique matricielle et d'optimisation, Dunod.

J-P. Demailly, Analyse numérique et équations différentielles, EDP Sciences.

M. Schatzman, Analyse numérique : une approche mathématique - Cours et exercices, Dunod.

J. Rappaz et M. Picasso, Introduction à l'analyse numérique, Presses polytechnique romandes.

P. Lascaux et R. Theodor, Analyse numérique matricielle appliquée à l'art de l'ingénieur - Méthodes directes, Dunod.

J-E. Rombaldi, Analyse matricielle - Cours et Exercices, EDP Sciences.

M. Sibony et J-C. Mardon, Analyse numérique - Tome I, Editions des sciences et des arts, Hermann.

M. Sibony et J-C. Mardon, Analyse numérique - Tome II, Editions des sciences et des arts, Hermann.

M. Sibony, Itérations et Approximations, Editions des sciences et des arts, Hermann.

Ressources web *Ci-joint une petite liste de site contenant des fiches d'analyse numérique.*

http ://lumimath.univ-mrs.fr/~jlm/cours/analnum/

http ://iacs.epfl.ch/asn/Support/support

http ://rfv.insa-lyon.fr/~jolion/ANUM/poly.html

http ://benallal.free.fr/an/

http ://www.unige.ch/~hairer/polycop.html

http ://www.cmi.univ-mrs.fr/~herbin/TELE/L3/

http ://www.mathappl.polymtl.ca/mth2210/biblionumer.html

http ://www.iecn.u-nancy.fr/~sokolows/support/support.html

http ://www.math-linux.com

Chapitre 1

Interpolation et Approximation

Soient $n+1$ couples : (x_0, f_0), (x_1, f_1), ..., $(x_n, f_n) \in \mathbf{R} \times \mathbf{R}$ tels que les x_i soient tous distincts. On cherche un polynôme $P : \mathbf{R} \longrightarrow \mathbf{R}$ tel que $P(x_i) = f_i$ pour $i = 0..n$.
On introduit dans un premier temps la *formule d'interpolation de Lagrange* qui prouve

l'existence et l'unicité du polynôme P de degré au plus n. Ensuite on présentera l'*algorithme d'Aitken* et la *méthode des différence divisées* qui utilise la *formule d'interpolation de Newton*. Puis, nous introduirons l'*algorithme de Neville* qui présente une méthode très pratique pour le calcul sur machine du polynôme P en un point. Nous introduirons enfin l'*interpolation Spline* qui est un outil fondamental du graphisme sur ordinateur.

1.1 Formule d'Interpolation de Lagrange

Généralement les réels f_i représentent le résultat d'expériences et sont l'expression d'une fonction f continue dans $\mathbf{R}$ en les points x_i. La formule d'interpolation de Lagrange est un moyen qui permet d'approcher la fonction f.

Théorème 1.1.1. *Soient (x_i, f_i) pour $i = 0..n$ des points réels ou complexes, avec $x_i \neq x_j$ pour $i \neq j$. Il existe un unique polynôme P de degré inférieur ou égal à n vérifiant $P(x_i) = f_i$ pour $i = 0..n$.*

La preuve du théorème 1.1.1 se base sur les *polynômes d'interpolation de Lagrange*

$$L_i(x) = \prod_{j=0, j\neq i}^{j=n} \frac{x - x_j}{x_i - x_j}$$

qui vérifient $L_i(x_i) = 1$ et pour $j \neq i$: $L_i(x_j) = 0$. Le polynôme P est alors égal à $P(x) = \sum_{i=0}^{i=n} f_i L_i(x)$ et il est appelé *polynôme ou formule de Lagrange.*

Démonstration : Soient P et Q deux polynômes de degré au plus égal à n et vérifiant $P(x_i) = Q(x_i) = f_i$ pour $i = 0..n$. Alors le polynôme $r = P - Q$ (qui est de degré au plus égal à n) s'annule en $x_0, x_1, \ldots, x_n$. C'est donc un polynôme qui a $n+1$ racines : il est nécessairement nul. Donc $P = Q$, d'où l'unicité.
Pour montrer l'existence d'un tel polynôme, il suffit de remarquer que la famille $(L_0, L_1, \ldots, L_n)$ forme une base de l'espace vectoriel des polynômes de degré inférieur ou égal à n (qui est

de dimension $n+1$). En effet, pour tout x, $\sum_{i=0}^{i=n} \alpha_i L_i(x) = 0$ implique en particulier que $\alpha_k = 0$ pour $k = 0..n$ (on remplace x par x_k). On retrouve la formule de Lagrange en écrivant P dans cette base : $P = \sum_{i=0}^{i=n} f_i L_i(x)$ et $P(x_i) = f_i$ pour tout $i = ..n$.

Exemple 1.1.1. Le polynôme P de degré inférieur ou égal à 3 vérifiant

$$P(0) = 1, \qquad P(1) = 2, \qquad P(-1) = -1 \qquad et \qquad P(2) = 4$$

est $P(x) = 1.L_0(x) + 2.L_1(x) + (-1).L_{-1}(x) + 4.L_2(x) = \frac{1}{3}x^3 - \frac{1}{2}x^2 + \frac{7}{6}x + 1$ où :

$$\begin{array}{lll} L_0(x) = & \frac{(x-1)(x+1)(x-2)}{(0-1)(0+1)(0-2)} & , \quad L_1(x) = \frac{(x-0)(x+1)(x-2)}{(1-0)(1+1)(1-2)}, \\ L_{-1}(x) = & \frac{(x-0)(x-1)(x-2)}{(-1-0)(-1-1)(-1-2)} & , \quad L_2(x) = \frac{(x-0)(x-1)(x+1)}{(2-0)(2-1)(2+1)}. \end{array}$$

Exercice 1.1.1. Construire les polynômes d'interpolation de Lagrange de $f(x) = e^{-x^2}$ sur les entiers de $[-2, 2]$.

Exercice 1.1.2. Ecrire un algorithme qui calcule $L_i(x)$ pour n, i et x fixés.

Remarque 1.1.1. Si on utilise la formule de Lagrange, le nombre d'opérations nécessaires pour calculer $P(x)$ est de l'ordre de $4n^2$ opérations pour chaque valeur de x.

Sur ordinateur, on préfère utiliser la *formule barycentrique* :

$$P(x) = \frac{\sum_{k=0}^{k=n} \frac{a_k f_k}{x - x_k}}{\sum_{k=0}^{k=n} \frac{a_k}{x - x_k}}$$

avec $a_k = \frac{1}{v'(x_k)}$ et $v(x) = (x - x_0)(x - x_1)\dots(x - x_n)$. Cette formule nécessite pour une première valeur de x l'ordre de $2n^2$ opérations afin de déterminer $P(x)$. Pour d'autres valeurs de x, $2n$ opérations sont nécessaire au calcul de P a condition de conserver les valeurs a_k.

1.2 Algorithme d'Aitken

Soit la relation de récurrence suivante :

$$\begin{cases} F_{k,0}(x) & = f_k, \qquad k = 0..n \\ F_{k,j+1}(x) & = \frac{F_{j,j}(x)(x_k - x) - F_{k,j}(x)(x_j - x)}{x_k - x_j}, \qquad k = j+1..n \end{cases}$$

Théorème 1.2.1. *Le polynôme de Lagrange $P(x)$ sur les points $x_0, x_1, \dots, x_n$ est égal à $F_{n,n}(x)$.*

La preuve du théorème 1.2.1 se base sur le fait que $F_{k,j}(x)$ avec $k \geq j$ est le polynôme de Lagrange sur les points $x_0, x_1, \dots, x_{j-1}, x_k$.

Algorithme. On calcule $P(x) = F_{n,n}(x)$ à l'aide du tableau triangulaire suivant :

	0	1	2	3	...	j	$j+1$	...	n	
0	$F_{0,0}$									$x_0 - x$
1	$F_{1,0}$	$F_{1,1}$								$x_1 - x$
2	$F_{2,0}$	$F_{2,1}$	$F_{2,2}$							$x_2 - x$
3	$F_{3,0}$	$F_{3,1}$	$F_{3,2}$	$F_{3,3}$						$x_3 - x$
$\vdots$					...					$\vdots$
k	$F_{k,0}$	$F_{k,1}$	$F_{k,2}$	$F_{k,3}$		$F_{k,j}$	$F_{k,j+1}$			$x_k - x$
$\vdots$										$\vdots$
n	$F_{n,0}$	$F_{n,1}$	$F_{n,2}$						$F_{n,n}$	$x_n - x$

Par exemple, pour calculer $F_{k,2}(x)$, on considère le produit croisé suivant : $F_{1,1}(x)(x_k - x) - F_{k,1}(x)(x_1 - x)$. Nous obtenons :

$$F_{k,2}(x) = \frac{F_{1,1}(x)(x_k - x) - F_{k,1}(x)(x_1 - x)}{x_k - x_1}.$$

Nous calcuons les éléments du tableau progressivement afin de déterminer $F_{n,n}(x) = P(x)$.

Démonstration du théorème 1.2.1 : Par récurrence sur j : $F_{k,0}(x) = f_k$ pour $k = 0..n$. Supposons que $F_{k,j}(x)$ est le polynôme de Lagrange sur les points $x_0, x_1, \ldots, x_{j-1}, x_k$ pour tout $k = j..n$.
En particulier, et d'après l'hypothèse de récurrence, $F_{j,j}(x)$ est le polynôme de Lagrange sur $x_0, x_1, \ldots, x_{j-1}, x_j$.
Montrons que $F_{k,j+1}(x)$ est le polynôme de Lagrange sur $x_0, x_1, \ldots, x_j, x_k$ pour $k = j+1..n$.
Pour $i = 0..j-1$:

$$F_{k,j+1}(x_i) = \frac{F_{j,j}(x_i)(x_k - x_i) - F_{k,j}(x_i)(x_j - x_i)}{x_k - x_j} = f_i$$

Nous utilisons pour cela l'hypothèse de récurrence : $F_{j,j}(x_i) = f_i$ et $F_{k,j}(x_i) = f_i$ pour $i = 0..j-1$). D'autre part, en utilisant la définition, on trouve que $F_{k,j+1}(x_j) = f_j$ et $F_{k,j+1}(x_k) = f_k$. En prenant $j = n$, on obtient le résultat du théorème.

Remarque 1.2.1. Le polynôme $F_{k,j}(x)$ avec $k \geq j$ est le polynôme de Lagrange sur les points $x_0, x_1, \ldots, x_{j-1}, x_k$ de degré inférieur ou égal à $j+1$.

Exemple 1.2.1. L'algorithme d'Aitken appliqué à l'exemple 1.1.1 donne :

	0	1	2	3	
0	1				$-x$
1	2	$F_{1,1} = 1 + x$			$1 - x$
2	-1	$F_{2,1} = 1$	$F_{2,2} = \frac{x^2}{2} - \frac{x}{2} + 1$		$-1 - x$
3	4	$F_{3,1} = 1 + \frac{3x}{2}$	$F_{3,2} = \frac{x^2}{2} + \frac{x}{2} + 1$	$F_{3,3} = \frac{x^3}{3} - \frac{x^2}{2} + \frac{7x}{6} + 1$	$2 - x$

1.2.1 Formule de Newton

Soient $n \in \mathbf{N}$ et $0 \leq i < j \leq n$ deux entiers distincts. Notons $D_{i,j}(x)$ le polynôme de Lagrange sur $x_i, x_{i+1}, \ldots, x_j$. C'est un polynôme de degré inférieur ou égal à $j-i$ vérifiant $D_{i,j}(x_k) = f_k$ pour $i \leq k \leq j$.

1.2.2 La méthode des différences divisées

Proposition 1.2.2 (relation des différences divisées)**.** *Notons $c_{i,j}$ le coefficient de x^{j-i} dans $D_{i,j}(x)$. Alors, on a la relation suivante :*

$$\begin{cases} c_{i,i} = f_i & \text{pour } i = 0..n, \\ c_{i,j} = \frac{c_{i+1,j}-c_{i,j-1}}{x_j-x_i} & \text{pour } 0 \leq i < j \leq n. \end{cases}$$

Démonstration : Soit $n \in \mathbf{N}$. Pour tout $0 \leq i < j \leq n$, on a

$$D_{i,j}(x) = \frac{D_{i+1,j}(x)(x-x_i) - D_{i,j-1}(x)(x-x_j)}{x_j - x_i} \quad . \tag{1.2.1}$$

En effet, le degré de $D_{i+1,j}(x)$ est $\leq j-i-1$ et le degré de $D_{i,j-1}(x)$ est $\leq j-i-1$ ce qui montre que le degré du polynôme $\frac{(x-x_i)D_{i+1,j}(x)-(x-x_j)D_{i,j-1}(x)}{x_j-x_i}$ est $\leq j-i$. De plus, $\frac{D_{i+1,j}(x_k)(x_k-x_i)-D_{i,j-1}(x_k)(x_k-x_j)}{x_j-x_i} = f_k$ (il faut distinguer les trois cas : $k=i$, $k=j$ et $k = i+1..j-1$ où $D_{i+1,j}(x_k) = D_{i,j-1}(x_k) = f_k$).

Le polynôme $\frac{D_{i+1,j}(x)(x-x_i)-D_{i,j-1}(x)(x-x_j)}{x_j-x_i}$ est alors le polynôme de Lagrange sur les points $x_i, \ldots, x_j$ et puisqu'il est unique (voir théorème 1.1.1) on a l'égalité (1.2.1) qui est équivalente à :

$$D_{i,j}(x) = \frac{D_{i+1,j}(x)(x-x_i) - D_{i,j-1}(x)(x-x_i)}{x_j - x_i} + D_{i,j-1}(x) \quad .$$

On remarque que $\frac{D_{i+1,j}(x)(x-x_i)-D_{i,j-1}(x)(x-x_i)}{x_j-x_i}$ est un polynôme de degré $\leq j-i$ qui s'annule en $x_i, x_{i+1}, ..x_{j-1}$ alors :

$$D_{i,j}(x) = c_{i,j}(x-x_i)(x-x_{i+1})\ldots(x-x_{j-1}) + D_{i,j-1}(x) \quad . \tag{1.2.2}$$

En examinant le coefficient de x^{j-i} de part et d'autre des deux égalités (1.2.1) et (1.2.2), nous obtenons la relation $c_{i,j} = \frac{c_{i+1,j}-c_{i,j-1}}{x_j-x_i}$:

$$c_{i,j}(x-x_i)(x-x_{i+1})\ldots(x-x_{j-1}) + D_{i,j-1}(x) = \frac{D_{i+1,j}(x)(x-x_i) - D_{i,j-1}(x)(x-x_j)}{x_j - x_i}.$$

En itérant le résultat de l'équation (1.2.2), on obtient :

$$D_{i,j}(x) = c_{i,j}(x-x_i)(x-x_{i+1})\ldots(x-x_{j-1}) + c_{i,j-1}(x-x_i)\ldots(x-x_{j-2}) + \cdots + c_{i,i}$$

et pour $i=0$ et $j=n$, on a :

Définition 1.2.1 (Formule de Newton)**.** Le polynôme de Lagrange $P(x)$ sur les points $x_0, x_1, \ldots, x_n$ est donné par la *formule de Newton* :

$$P(x) = D_{0,n}(x) = f_0 + \sum_{k=1}^{k=n} c_{0,k}(x-x_0)(x-x_1)\ldots(x-x_{k-1}) \quad .$$

$c_{0,k}$ est le coefficient de x^k appelée *la différence divisée d'ordre k aux points $x_0, x_1, \ldots, x_k$.*

La *formule de Newton* permet de déterminer le polynôme de Lagrange. Pour cela, nous utilisons un tableau analogue à la méthode d'Aitken : les coefficients de $P(x)$ sont sur la diagonale du tableau triangulaire.

Exemple 1.2.2. En utilisant la méthode des différences divisées et plus précisément la proposition 1.2.2 dans l'exemple 1.1.1 on obtient :

	x_j	$i=j$	$i=j-1$	$i=j-2$	$i=j-3$
$j=0$	0	$c_{0,0}=1$			
$j=1$	1	$c_{1,1}=2$	$c_{0,1}=1$		
$j=2$	-1	$c_{2,2}=-1$	$c_{1,2}=\frac{3}{2}$	$c_{0,2}=-\frac{1}{2}$	
$j=3$	2	$c_{3,3}=4$	$c_{2,3}=\frac{5}{3}$	$c_{1,3}=\frac{1}{6}$	$c_{0,3}=\frac{1}{3}$

Nous remplissons le tableau ligne par ligne avec $c_{i,i}=f_i$ pour $i=0..n$ et $c_{i,j}=\frac{c_{i+1,j}-c_{i,j-1}}{x_j-x_i}$ pour $0\leq i<j\leq 3$. Ainsi, $D_{0,3}(x)=\frac{1}{3}(x-0)(x-1)(x+1)+(-\frac{1}{2})(x-0)(x-1)+x+1=\frac{1}{3}x^3-\frac{1}{2}x^2+\frac{7}{6}x+1$.

1.2.3 Algorithme de Neville

L'algorithme de calcul du polynôme de Lagrange $P(x)=D_{0,n}(x)$ pour x fixé est constitué des n étapes suivantes :

Etape 1 Pour i allant de 0 à $n-1$, calculer

$$D_{i,i+1}(x)=\frac{f_{i+1}(x-x_i)-f_i(x-x_{i+1})}{x_{i+1}-x_i}$$

Etape 2 Pour i allant de 0 à $n-2$, calculer

$$D_{i,i+2}(x)=\frac{D_{i+1,i+2}(x)(x-x_i)-D_{i,i+1}(x)(x-x_{i+2})}{x_{i+2}-x_i}$$

$\vdots$

Etape $n-1$ Pour i allant de 0 à 1, calculer

$$D_{i,i+n-1}(x)=\frac{D_{i+1,i+n-1}(x)(x-x_i)-D_{i,i+n-2}(x)(x-x_{i+n-1})}{x_{i+n-1}-x_i}$$

Etape n Pour $i=0$ Calculer $D_{i,i+n}(x)=P(x)$.

1.3 Estimation de l'erreur

Pour évaluer l'erreur d'interpolation de Lagrange, on a le théorème suivant :

Théorème 1.3.1. *Soient f une fonction de classe C^{n+1} sur un intervalle $[a,b]$ et P le polynôme de Lagrange de f en les points $x_0<x_1<\cdots<x_n\in[a,b]$. Alors*

$$f(x)-P(x)=\frac{(x-x_0)(x-x_1)\dots(x-x_n)}{(n+1)!}\,f^{(n+1)}(\zeta) \tag{1.3.1}$$

où $a\leq min(x,x_0)<\zeta<max(x,x_n)\leq b$.

Démonstration : La relation est triviale pour $x = x_i \ \forall i = 0..n$. Soit $x \neq x_i$, pour tout $i = 0..n$. On pose $k(x) = \frac{f(x)-P(x)}{(x-x_0)(x-x_1)...(x-x_n)}$ et $w(t)$ un polynôme qui s'annule en x_0, x_1, ..., x_n, et x définit par : $w(t) = f(t) - P(t) - (t-x_0)(t-x_1)\dots(t-x_n)k(x)$. Il s'annule en $(n+2)$ points. En appliquant le thórème de Rolle, on montre que $w^{n+1}(t)$ s'annule en au moins un point ζ :

$$w^{n+1}(\zeta) = f^{n+1}(\zeta) - (n+1)!\, k(x) = 0 \qquad et \qquad k(x) = \frac{f^{n+1}(\zeta)}{(n+1)!} \qquad .$$

On remarque que l'erreur dépend de la fonction $f^{n+1}(\zeta)$ et des points d'interpolation par $(x-x_0)(x-x_1)\dots(x-x_n)$. Considérons le cas où les points d'interpolation sont équidistants, c'est-à-dire, $x_i = a + hi$ pour $i = 0..n$ avec $h = \frac{b-a}{n}$. Si f est continue et que toutes ses dérivées sont continues jusqu'à l'ordre $n+1$, alors

$$\mid f(x) - P(x) \mid \leq \frac{1}{2(n+1)}(\frac{b-a}{n})^{n+1}\ max_{a\leq x\leq b} \mid f^{(n+1)}(x) \mid \qquad (1.3.2)$$

Cette quantité ne tend pas nécessairement vers 0 car les dérivées $f^{(n+1)}$ de f peuvent grandir très vite lorsque n croît. Dans l'exemple suivant, nous présentons deux cas sur la convergence de l'interpolation de Lagrange.

Exemple 1.3.1. Soit $f(x) = \sin(x)$, $\mid f^k(x) \mid \leq 1$: l'interpolé P converge vers f quelque soit le nombre de points d'interpolation et leur emplacement. Par contre, pour $f(x) = \frac{1}{1+25x^2}$ sur $[-1,1]$ et bien que f soit indéfiniment continûment dérivable sur $[-1,1]$, les grandeurs

$$max_{-1\leq x\leq 1} \mid f^k(x) \mid \qquad , \qquad k = 1,2,3,\dots$$

explosent très rapidement et l'inégalité (1.3.2) ne nous assure plus la convergence de l'interpolation.

Le choix des points x_i apporte une amélioration sensible de l'interpolation : Lorsque $a \leq x \leq b$, on choisit les abscisses d'interpolation $x_k = \frac{a+b}{2} + \frac{b-a}{2}\cos(\frac{2k+1}{2(n+1)}\pi)$ pour $k = 0..n$, on obtient :

$$\mid f(x) - P(x) \mid \leq \frac{(b-a)^{n+1}}{(n+1)!2^{2n+1}}\ max_{x\in[a,b]} \mid f^{(n+1)}(x) \mid \qquad .$$

Les points $x_k = \frac{a+b}{2} + \frac{b-a}{2}\cos(\frac{2k+1}{2(n+1)}\pi)$ pour $k = 0..n$ sont calculés à partir des zéros des polynômes de Tchebycheff ($T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$).

Nous venons de voir que l'interpolant de Lagrange peut diverger lorsque le nombre de points d'interpolation n devient grand et lorsque les points d'interpolation sont équidistants. Une première solution consiste à découper l'intervalle $[a,b]$ en plusieurs sous-intervalles, c'est l'interpolation par intervalles. On peut également utiliser des points judicieusement placés (zéros des polynômes de Tchebycheff par exemple).

1.4 Interpolation Spline

Nous rencontrons les splines dans la majorité des programmes de CAO et il est certainement bon pour un ingénieur d'en connaître l'existence et les principes.

L'interpolation spline est une approche locale en faisant de l'interpolation par morceaux : sur un sous-intervalle donné, on fait de l'interpolation polynomiale de degré faible (≤ 3). Les splines réalisent une interpolation polynomiale par morceaux et on imposera aux noeuds un degré de régularité suffisant. Avec les splines, on obtient une très bonne approximation, sans effets de bord, contrairement à l'interpolation de Lagrange.

Etant donnée une fonction f définie sur $[a, b]$ et un ensemble de noeuds $a \leq x_0 < x_1 < \cdots < x_n \leq b$. Nous appelons *spline d'ordre* 3 ou *spline cubique* interpolant f aux noeuds, une fonction S vérifiant les conditions suivantes :

1. S coïncide avec un polynôme P_i de degré 3 sur chacun des intervalles $[x_{i-1}, x_i]$ pour $i = 1..n$.
2. $S(x_i) = P_i(x_i) = f_i$ pour $i = 0..n$.
3. $P_i(x_i) = P_{i+1}(x_i)$ pour $i = 0..n-1$, (continuité de S).
4. $P_i^{'}(x_i) = P_{i+1}^{'}(x_i)$ pour $i = 0..n-1$, (continuité de $S^{'}$).
5. $P_i^{''}(x_i) = P_{i+1}^{''}(x_i)$ pour $i = 0..n-1$, (continuité de $S^{''}$).
6. S satisfait une des conditions au bord suivantes :
   $S^{''}(x_0) = S^{''}(x_n) = 0$ (spline libre) ou bien $S^{'}(x_0) = f_0^{'}$ et $S^{'}(x_n) = f_n^{'}$ (spline forcée lorsque f est dérivable).

Proposition 1.4.1. *Les polynômes $P_i(x)$ sont données par les formules suivantes :*

$$P_i(x) = M_{i-1}\frac{(x_i - x)((x_i - x)^2 - h_i^2)}{6h_i} + M_i\frac{(x - x_{i-1})((x - x_{i-1}^2) - h_i^2)}{6h_i} + \frac{f_{i-1}(x_i - x)}{h_i} + \frac{f_i(x - x_{i-1})}{h_i}$$

avec $h_i = x_i - x_{i-1}$ pour $i = 1..n$, $M_0 = M_n = 0$ et $M_1, M_2, \ldots, M_{n-1}$ sont solutions du système linéaire :

$$\frac{h_i}{6}M_{i-1} + \frac{h_i + h_{i+1}}{3}M_i + \frac{h_{i+1}}{6}M_{i+1} = \frac{f_{i+1} - f_i}{h_{i+1}} - \frac{f_i - f_{i-1}}{h_i} \qquad \text{pour } i = 1..n-1.$$

Indications sur la preuve : Pour montrer ces égalités, on commence par exprimer $P_i^{''}$ en fonction de $f_i^{''}$ et $f_{i+1}^{''}$ par une interpolation linéaire : $P_i^{''}(x) = f_i^{''}\frac{x_{i+1}-x}{x_{i+1}-x_i} + f_{i+1}^{''}\frac{x-x_i}{x_{i+1}-x_i}$. Puis, on intègre deux fois. En tenant compte des raccords aux noeuds, on obtient un système linéaire de $(n-1)$ équations et $(n+1)$ inconnues.

Exemple 1.4.1. On reprend l'exemple 1.1.1 de la section précédente. On cherche la spline cubique $S(x)$ telle que $S(0) = 1$, $S(1) = 2$, $S(-1) = -1$ et $S(2) = 4$. On pose $x_0 = -1$, $x_1 = 0$, $x_2 = 1$ et $x_3 = 2$. Ils sont bien ordonnées. Alors $f_0 = -1$, $f_1 = 1$, $f_2 = 2$, $f_3 = 4$ et $h_i = 1$ pour $i = 1..3$. On prend $M_0 = M_3 = 0$; M_1 et M_2 sont solution du système :

$$\begin{cases} \frac{2}{3} & M_1 & + & \frac{1}{6} & M_2 & = & -1 \\ \\ \frac{1}{6} & M_1 & + & \frac{2}{3} & M_2 & = & 1 \end{cases}$$

Nous obtenons :

$$S(x) = \begin{cases} P_1(x) = & -\frac{1}{3}x^3 - x^2 + \frac{4}{3}x + 1 & x \in [-1, 0] \\ P_2(x) = & \frac{2}{3}x^3 - x^2 + \frac{4}{3}x + 1 & x \in [0, 1] \\ P_3(x) = & -\frac{1}{3}x^3 + 2x^2 - \frac{5}{3}x + 2 & x \in [1, 2] \end{cases}$$

1.5 Introduction à l'Approximation

Soit f une fonction réelle définie soit de façon discrète, soit de façon continue. Le problème général de l'approximation consiste à déterminer une fonction g de $\mathbf{R}$ dans $\mathbf{R}$ de forme donnée, qui, en un certain sens, approche le mieux possible la fonction f. Un cas particulier est l'interpolation polynomiale (la fonction g est alors un polynôme). Mais l'interpolation est un outil de peu de valeur pour le traitement des données expérimentales. On l'utilise seulement lorsque les données sont très précises. Cette section traite particulièrement de la *méthode des moindres carrés discrets*.

1.5.1 Résultats fondamentaux

Il est relativement fréquent d'avoir à ajuster sur des données une loi empirique ou encore d'avoir à déterminer la valeur des paramètres d'une loi théorique de façon à reproduire les résultats d'une expérience. Une façon possible pour aborder de tels problèmes est de chercher les valeurs de paramètres minimisant un critère.

Soient u_0, u_1, ..., u_p, $p+1$ élémets d'un espace vectoriel normé E. Pour tout $y \in E$, le problème de la meilleure approximation de y admet une solution, c'est-à-dire que la fonction *norme de l'erreur* $\epsilon(a_0, \ldots, a_n) = \| y - \sum_{i=0}^{n} a_i u_i \|^2$ admet un minimum a dans E. Pour cela, on montre que cette fonction est continue sur un fermé borné : elle atteint donc son minimum.

Notons V le sous-espace vectoriel engendré par les $p+1$ points $u_0, u_1, \ldots, u_p$ de E deux à deux distincts. La meilleure approximation de y sur V au sens des moindres carrés est $\sum_{i=0}^{i=p} a_i u_i$ où les a_i, $i = 0..p$ sont solutions du système linéaire :

$$\sum_{i=0}^{i=p} a_i \ < u_i, u_j > = < y, u_j > \qquad pour \qquad j = 0..p. \tag{1.5.1}$$

1.5.2 Approximation polynomiale discrète

Il y a une manière assez simple de trouver l'équation d'une courbe. On positionne les données sur un graphique, on deplace la règle jusqu'à ce que la ligne se trouve équidistante de tous les points. L'analyse numérique change cette supposition à vue d'oeil en un processus plus scientifique c'est l'approximation au sens des moindres carrés.

Notons $E = \mathbf{R}[x]$ l'espace vectoriel des polynômes en la variable x et à coefficient dans $\mathbf{R}$. Soit V l'espace vectoriel des polynômes de degré $\leq p$. On pose $u_0 = x^p$, $u_1 = x^{p-1}, \ldots, u_{p-1} = x$ et $u_p = 1$.

On considère les abscisses $x_0, x_1, \ldots, x_n \in \mathbf{R}$ et f une fonction de $\mathbf{R}$ dans $\mathbf{R}$ telle que $f(x_i) = y_i$ pour $i = 0..n$. On veut approcher l'ensemble discret des points $(x_i, y_i)_{0 \leq i \leq n}$ par un polynôme de degré p à coefficient dans $\mathbf{R}$:

$$P_p(x) = a_0 x^p + a_1 x^{p-1} + \cdots + a_{p-1} x + a_p \qquad .$$

Pour tout $g_1, g_2 \in V$, on définit le produit scalaire sur V par

$$< g_1(x), g_2(x) > = \sum_{i=0}^{i=n} g_1(x_i) g_2(x_i) \ .$$

Le système (1.5.1) est alors équivalent au système suivant :

$$\begin{pmatrix} \sum_{i=0}^{i=n} x_i^{2p} & \sum_{i=0}^{i=n} x_i^{2p-1} & \sum_{i=0}^{i=n} x_i^{2p-2} & \cdots & \cdots & \sum_{i=0}^{i=n} x_i^{p} \\ \sum_{i=0}^{i=n} x_i^{2p-1} & \sum_{i=0}^{i=n} x_i^{2p-2} & \cdots & \cdots & \cdots & \sum_{i=0}^{i=n} x_i^{p-1} \\ \sum_{i=0}^{i=n} x_i^{2p-2} & & & & & \vdots \\ \vdots & & & & & \vdots \\ \vdots & & & & & \sum_{i=0}^{i=n} x_i \\ \sum_{i=0}^{i=n} x_i^{p} & \sum_{i=0}^{i=n} x_i^{p-1} & \cdots & \cdots & \sum_{i=0}^{i=n} x_i & p+1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ \vdots \\ \vdots \\ a_p \end{pmatrix} = \begin{pmatrix} \sum y_i x_i^p \\ \sum y_i x_i^{p-1} \\ \vdots \\ \vdots \\ \sum y_i x_i \\ \sum y_i \end{pmatrix}$$

Lorsque $p = 1$, on parle de *régression* et dans ce cas, on cherche une droite $P_1(x) = a_0x+a_1$ tel que

$$\sum_{i=0}^{i=n}(y_i - a_0x_i - a_1)^2 \leq \sum_{i=0}^{i=n}(y_i - \alpha_0x_i - \alpha_1)^2 \qquad \forall \alpha_0, \alpha_1 \in \mathbf{R} \qquad .$$

Le système (1.5.1) est alors équivalent à :

$$\begin{pmatrix} \sum_{i=0}^{i=n} x_i^2 & \sum_{i=0}^{i=n} x_i \\ \sum_{i=0}^{i=n} x_i & n+1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^{i=n} x_iy_i \\ \sum_{i=0}^{i=n} y_i \end{pmatrix} \qquad . \tag{1.5.2}$$

En posant $X = (x_0, x_1, \ldots, x_n)$ et $Y = (y_0, y_1, \ldots, y_n)$, on obtient à l'aide des formules de la covariance et de la moyenne :

$$\begin{array}{rcl} a_0 & = & \frac{\sigma(X,Y)}{\sigma(X^2)} \qquad avec \quad \sigma(X,Y) = \frac{1}{n+1}\sum_{i=0}^{i=n}(x_i - \bar{X})(y_i - \bar{Y}) \\ a_1 & = & \bar{Y} - a_0\bar{X} \quad et \qquad \bar{X} = \frac{1}{n+1}\sum_{i=0}^{i=n} x_i \end{array}$$

Le point moyen $(\bar{X}, \bar{Y})$ appartient à la droite d'équation $P_1(x) = a_0x + a_1$.

Exemple 1.5.1. On considère la fonction $f(x) = x^2 + 2x$ sur $[-1, 1]$. Calculer les valeurs de f aux points d'abscisses $-1, -\frac{1}{2}, 0, \frac{1}{2}$ et 1. Donner la droite de meilleure approximation discrète de f pour ces 5 points au sens des moindres carrés. On note $x_0 = -1$, $x_1 = -\frac{1}{2}$, $x_2 = 0$, $x_3 = \frac{1}{2}$ et $x_4 = 1$. Alors $y_0 = -1$, $y_1 = -\frac{3}{4}$, $y_2 = 0$, $y_3 = \frac{5}{4}$ et $y_4 = 3$. On calcule $\sum_{i=0}^{i=4} x_i^2 = \frac{5}{2}$, $\sum_{i=0}^{i=4} x_i = 0$, $\sum_{i=0}^{i=4} x_iy_x = 5$ et $\sum_{i=0}^{i=4} y_i = \frac{5}{2}$. On obtient le système suivant à partir du système (1.5.2) :

$$\begin{pmatrix} \sum_{i=0}^{i=4} x_i^2 = \frac{5}{2} & \sum_{i=0}^{i=n} x_i = 0 \\ \sum_{i=0}^{i=n} x_i = 0 & n+1 = 5 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^{i=n} x_iy_i = 5 \\ \sum_{i=0}^{i=n} y_i = \frac{5}{2} \end{pmatrix} \qquad .$$

et $a_0 = 2$, $a_1 = \frac{1}{2}$. Donc $P_1(x) = 2x + \frac{1}{2}$.

1.6 Exercices d'Applications

1. Implémenter l'algorithme de Neville dans le logiciel de Calcul Formel Maple pour x fixé.
2. Construire le polynôme de Lagrange de $f(x) = e^{-x^2}$ sur les entiers de $[-2, 2]$ par la méthode des différences divisées.
3. Soit la fonction f de $\mathbf{R}$ dans $\mathbf{R}$ définie par $f(x) = \frac{1}{1+x^2}$.

(a) Construire le polynôme de Lagrange P de f par la méthode des différences divisées sur 0, 1, 3 et 5.

(b) Comparer $P(4)$ et $f(4)$.

(c) Que peut-on conclure ?

4. On recueille les données suivantes concernant la météo dans la région de Bizerte le 13 janvier 2003 :

temps (heures)	0h00	5h00	7H00	12h00
pluviométrie (millimètres)	10	3	5	2

(a) Utiliser la méthode d'Aitken pour calculer le polynôme interpolant la fonction "pluviométrie".

(b) Approximer la pluviométrie dans la région de Bizerte au temps t=10h00.

5. Soit $f(x) = x + exp(-x^2)$. Donner la formule du polynôme d'interpolation de Lagrange $L_i(x)$ sur les points $(i, f(i))$ pour i fixé.

6. Poser $f(x) = \exp(-x^2)$, $a = -1$, $b = 2$ et

$$\mathbf{x}_k = \frac{a+b}{2} + \frac{b-a}{2}\cos(\frac{2k+1}{2(n+1)}\pi)$$

pour k allant de 1 à 3.

(a) Tracer l'ensemble des zéros du polynômes de Tchebycheff sur un même graphe que la fonction f.

(b) Calculer le polynôme de Lagrange en ces points et comparer avec la fonction f.

7. Déterminer la spline cubique des points $(-1, 0)$, $(0, 1)$, $(1, 3)$ et $(2, 2)$. Comparer cette spline avec le polynôme de Lagrange en ces même points.

8. L'évolution de la concentration c d'une espèce dans un mélange est donnée par une loi linéaire en fonction du temps : $c(t) = a_0 t + a_1$. Un expérimentateur fait une série de mesures pour déterminer les paramètres inconnus :

0	1	2	3	4	5	6	7	8	9	10	11
1,54	3,09	4,97	7,43	10,65	14,92	20,6	28,2	38,42	52,15	70,65	96,6

(a) On pose $n = 11$. Calculer $\sum_{i=0}^{i=n} t_i^2$, $\sum_{i=0}^{i=n} t_i$, $\sum_{i=0}^{i=n} t_i c_i$ et $\sum_{i=0}^{i=n} c_i$ et écrire le système pour l'approximation au sens des moindres carrées.

(b) Utiliser la méthode de régression des moindres carrés pour obtenir les paramètres a_0 et a_1. Rappelons que

$$\mathbf{a}_0 = \frac{\sigma(X,Y)}{\sigma(X^2)} \; et \; \mathbf{a}_1 = \bar{Y} - a_0\bar{X} \; .$$

9. (a) Trouver le polyôme P du secons degré passant par $(0, 0)$, $(1, 1)$, $(2, 8)$.

(b) La fonction x^3 prend les valeurs de P. Utiliser la formule vue en cours pour exprimer l'erreur d'interpolation commise en remplaçant x^3 par le polynôme d'interpolation P.

10. (a) Interpoler e^x aux points 0, $\frac{1}{2}$ et 1 par un polynôme du second degré.

(b) Trouver une expression pour l'erreur d'interpolation et une borne de l'erreur indépendante de ζ.

11. (a) Sachant que la température initiale d'un objet est de $0^{\circ}\mathbf{C}$, puis de $5^{\circ}\mathbf{C}$ après une minute et $3^{\circ}\mathbf{C}$ après deux minutes, quelle a été la température maximale et quant a-t-elle été atteinte ?

    (b) Quelle est la température moyenne entre 1 et 2 minutes ?

12. Considérons l'ensemble des points

$$\begin{array}{llll} x_0 = -1, & x_1 = 0, & x_2 = 1, & x_3 = 2, \\ f_0 = 7, & f_1 = 2, & f_2 = -1, & f_3 = -14. \end{array}$$

    (a) Calculer les polynômes d'interpolation de Lagrange L_0 et L_3.

    (b) Vérifier que $L_i(x_i) = 1$ et que $L_i(x_j) \neq 0$ pour tout i et $j \neq i$ de 0 à 3.

    (c) Citer deux méthodes permettant de calculer le polynôme de Lagrange.

    (d) Vérifier que le polynôme de Lagrange sur les points (x_0, f_0), (x_1, f_1), (x_2, f_2) et (x_3, f_3) est égal à :

$$P(x) = -2x^3 + x^2 - 2x + 2 \qquad .$$

    (e) Afin de déterminer une approximation du nuage de points (x_i, f_i) pour i de 0 à 3, calculer la droite de régression linéaire P_0 par la méthode des moindres carrés discrets.

    (f) Quelle est la différence entre P et P_0.

Méthode d'Hermite. Soit $x_0, x_1, \ldots, x_n$, $(n+1)$ points de a, b et f une fonction de classe C^1 sur a, b. Au lieu de chercher à faire coïncider f et P_n aux points x_i, nous pouvons aussi chercher à faire coïncider f et P_n ainsi que leurs dérivées jusqu'à un ordre donné aux points x_i.
De la même manière que dans l'interpolation de Lagrange, nous cherchons un polynôme P_n tel que $P_n(x_i) = f(x_i)$ et $P_n'(x_i) = f'(x_i)$.

Chapitre 2

Racine d'équation non linéaire

Les méthodes analytiques de résolution d'équations algébriques polynomiales sont limités à certaines formes et à de faible degré (équations quadratiques, cubiques, quartiques, $ax^{2n} + bx^n + c = 0$, etc...). La résolution des équations polynomiales de degré 4 est déjà plus pénible et à partir du degré 5, E. Galois a montré qu'il n'est pas toujours possible de donner une forme exacte des racines au moyen des coefficients (on utilise alors le groupe de l'équation, l'ideal des relations entre les racines ou encore le corps de décomposition du polynôme : ce sont des méthodes formelles [**?**]).

Pour la résolution des équations polynomiales, nous sommes amenés à l'utilisation des méthodes numériques. Il en est de même pour les équations contenant des fonctions transcendantes (i.e. non-algébriques) telles que $\exp(x)$, $\mathrm{ch}(x)$, etc... En étudiant les méthodes de résolution d'une équation f, on se posera deux questions :

1. La méthode converge-t-elle vers x^* où $f(x^*) = 0$. En d'autre termes la suite $(x_n)_{n\in\mathbf{N}}$ générée par la méthode converge-t-elle vers la solution x^*.
2. Dans l'affirmative, avec quelle rapidité (ceci permet de comparer les méthodes entre-elles).

2.1 Quelques algorithmes classiques

On désigne par (x_n) une suite de nombres réels $x_0, x_1, \ldots, x_n, \ldots$. Si une telle suite converge vers un point x^*, on écrira $(x_n) \longrightarrow x^*$. En analyse numérique, on construit les suites à l'aide d'algorithmes.

2.1.1 Méthode de la bissection

On considère une fonction f continue et on cherche x^* solution de $f(x) = 0$. On suppose qu'on a localisé par tâtonnement un intervalle $[a, b]$ dans lequel la fonction f change de signe : $f(a).f(b) < 0$.

On est donc certain qu'il y a entre a et b au moins un zéro de f. Pour approcher de façon précise l'un de ces zéros, on utilise l'algorithme suivant :

Algorithme 1 (Dichotomie).

Entrée : les extrémitées **a** et **b**,
la fonction **f**, la précision ϵ souhaitée et
N le nombre d'itérations maximum.
Sortie : Une approximation c de la racine x^* de **f**
ou bien un message d'erreur.
1. **Si** $f(a).f(b) > 0$ **Alors** Imprimer(*pas de solution*).
Sinon
$n := 1$;
2. **Tant que** $n \leq N$ **Faire**
3. $c := \frac{a+b}{2}$;
4. **Si** $f(c) = 0$ **ou** $\frac{b-a}{2} \leq \epsilon$ **Alors**
5. Imprimer(c) ; $n := N + 2$;
Fin Si ;
6. $n := n + 1$;
7. **Si** $f(a).f(c) > 0$ **Alors** $a := c$; **Sinon** $b := c$;
Fin Tant que ;
Si $n = N + 2$ **Alors** Imprimer(c)
Sinon Imprimer(*Après N itérations l'approximation de* x^*
obtenue est c et l'erreur maximale est $\frac{b-a}{2}$) ;
Fin Si ;
Fin Si ;
Fin.

A chaque itération, l'algorithme construit un nouvel intervalle, autour de x^*, qui est de longueur égale à la moitié de la longueur de l'intervalle précédent. Au bout de n itérations, la longueur de l'intervalle est alors de $\frac{b-a}{2^n}$.

Le nombre d'itérations nécessaire pour obtenir une précision de calcul égal à ϵ est définie par :

$$n \geq \frac{\log(b-a) - \log\epsilon}{\log 2} \qquad .$$

Remarque 2.1.1. Les avantages de cet algorithme sont la convergence assurée et une marge d'erreur sûre. Par contre il est relativement lent. Il est souvent utilisé pour déterminer une approximation initiale à utiliser avec un algorithme plus rapide.

2.1.2 Méthode de Newton

Cet algorithme se base sur la suite suivante (donner une explication géométrique) :

$$\begin{cases} x_0 & \text{donné} \\ x_{n+1} = & x_n - \frac{f(x_n)}{f'(x_n)}. \end{cases}$$

La signification géométrique consiste à mener par le point $(x_n, f(x_n))$ la tangente à la courbe $f(x)$. La pente de cette tangente est

$$T(x) = f(x_n) + (x - x_n)f'(x_n) \qquad .$$

Si on cherche le point d'intersection de la tangente avec l'axe des x, i.e. si on résoud $T(x) = 0$, on retrouve x_{n+1} tel que défini par la suite.

Théorème 2.1.1. *[Convergence globale de la méthode de Newton] Soit f une fonction de Classe C^2 sur $[a,b]$ et $g(x) = x - \frac{f(x)}{f'(x)}$. Si f vérifie :*

1. $f(a).f(b) < 0$
2. $\forall x \in [a,b]$, $f'(x) \neq 0$ *(c'est la stricte monotonie),*
3. $\forall x \in [a,b]$, $f''(x) \neq 0$ *(concavité dans le même sens).*

Alors en choisissant $x_0 \in [a,b]$ tel que $f(x_0).f''(x_0) > 0$, la suite (x_n) définie par x_0 et $x_{n+1} = g(x_n)$ converge vers l'unique solution de $f(x) = 0$ dans $[a,b]$.

Preuve : Les deux premières conditions du théorème assurent l'existence et l'unicité d'une racine x^* unique de f dans $[a,b]$. D'autre part, comme $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ on obtient :

$$\begin{aligned} x_{n+1} - x^* &= x_n - x^* - \frac{f(x_n)}{f'(x_n)} \\ &= x_n - x^* + \frac{f(x^*) - f(x_n)}{f'(x_n)} \\ &= x_n - x^* + \frac{(x^* - x_n)f'(x_n) + \frac{(x^*-x_n)^2}{2} f''(\zeta_n)}{f'(x_n)} \end{aligned}$$

Ainsi :

$$\begin{aligned} x_{n+1} - x^* &= (x_n - x^*)(1 - \frac{f'(x_n) + \frac{x^*-x_n}{2} f''(\zeta_n)}{f'(x_n)}) \\ &= \frac{(x_n - x^*)^2}{2} \frac{f''(\zeta_n)}{f'(x_n)} \end{aligned}$$

Si f'' et f' sont de même signe, alors $x_{n+1} - x^* > 0$ et la suite est alors minorée par x^* à partir du rang 1. Sinon, si f'' et f' sont de signes contraires, alors $x_{n+1} - x^* < 0$ et la suite est alors majorée. De plus, et selon les cas, la suite va être soit décroissante ($f'' > 0$ et $f' > 0$ ou bien $f'' < 0$ et $f' < 0$) donc convergente soit croissante ($f'' > 0$ et $f' < 0$ ou bien $f'' < 0$ et $f' > 0$) et donc convergente.

Algorithme 2 (Newton).

Entrée :	Une approximation initiale x_0, la précision ϵ souhaitée
	et **N** le nombre maximum d'itérations.
Sortie :	Une approximation c de la racine x^* de f ou bien un message d'erreur.
1.	$n := 1$;
2.	**Tant que** $n \leq N$ **Faire**
	$c := x_0 - \frac{f(x_0)}{f'(x_0)}$;
3.	**Si** $\mid c - x_0 \mid \leq \epsilon$ **Alors** c ; $n := N + 2$ **Fin Si** ;
	$n := n + 1$;
	$x_0 := c$;
	Fin Tant que ;
	Si $n = N + 2$ **Alors** Imprimer(c) **Sinon**
	Imprimer(*La méthode a échoué après N itérations*) ;
	Fin Si ;
Fin.	

Remarque 2.1.2. Il est essentiel de fixer une limite au nombre d'itérations car la convergence n'est pas assurée. On verra dans la suite qu'on peut construire des exemples pour lesquels cet algorithme ne converge pas. Cependant lorsque la convergence aura lieu, elle sera trés rapide comme le montre le théorème 2.1.2. D'autre part, le test d'arrêt n'assure pas que la racine x^* soit à une distance ϵ de x.

Définition 2.1.1. La méthode définie par $x_{n+1} = g(x_n)$ est dite *d'ordre p* si la limite de $|\ \frac{e_{n+1}}{e_n^p}\ |$ lorsque $n \longrightarrow +\infty$ est une constante réelle strictement positive.

Théorème 2.1.2. *Soit f une fonction de Classe C^2. Si x^* est une racine simple de f, alors la méthode de Newton est au moins d'ordre 2.*

Preuve : On remarque que $g'(x) = \frac{f(x).f''(x)}{(f'(x))^2}$ donc $g'(x^*) = 0$. D'autre part :

$$\begin{aligned} x_{n+1} - x^* &= g(x_n) - g(x^*) \\ &= (x_n - x^*)^2 \frac{g''(x^*)}{2} + (x_n - x^*)^2 \epsilon(e_n) \\ &= e_n^2 \frac{g''(x^*)}{2} + e_n^2 \epsilon(e_n) \end{aligned}$$

Comme $\epsilon(e_n)$ est négligeable au voisinage de $+\infty$ on obtient le résultat. A chaque itération, on multiplie par 2 le nombre de chiffres exacts de l'approximation.
D'une façon générale, si f est de Classe C^p alors : $e_{n+1} = e_n g'(x^*) + \frac{e_n^2}{2!} g''(x^*) + \cdots + \frac{e_n^p}{p!} g^{(p)}(x^*) + e_n^p \epsilon(e_n)$. La méthode est alors d'ordre p si $g'(x^*) = \cdots = g^{(p-1)}(x^*) = 0$ et $g^{(p)}(x^*) \neq 0$.

Il est parfois ennuyeux dans l'utilisation de la méthode de Newton-Raphson d'avoir à calculer $f^{'}(x)$. Il se peut par exemple qu'on ne dispose pas du programme d'ordinateur pour en effectuer le calcul alors qu'on peut facilement calculer $f(x)$. L'algorithme suivant peut être considéré comme une approximation de la méthode de Newton.

2.1.3 Méthode de la sécante

Au lieu d'utiliser la tangente au point x_n, nous allons utiliser la sécante passant par les points d'abscisses x_n et x_{n-1} pour en déduire x_{n+1} (donner une explication géométrique). L'équation de la sécante s'écrit sous la forme :

$$S(x) = f(x_n) + (x - x_n)\tau_n \qquad avec \qquad \tau_n = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}} \quad .$$

On définit x_{n+1} comme étant l'intersection de la sécante $S(x)$ avec l'axe des x : $S(x_{n+1}) = 0$ et on obtient

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

Remarque 2.1.3. La méthode de la sécante est une méthode itérative à deux pas en ce sens que x_{n+1} dépend de x_n et de x_{n-1}. Dans la méthode de Newton-Raphson x_{n+1} ne dépend que de x_n. C'est une méthode à un pas.

Une façon commode de procéder est de choisir x_0 arbitrairement puis de choisir $x_1 = x_0 + h$ où h est un accroissement petit. Si x_n et x_{n+1} sont proches (on peut espérer que ce sera le cas si la méthode converge), on peut considérer que $\frac{f(x_n)-f(x_{n-1})}{x_n-x_{n-1}}$ est une approximation de $f'(x_n)$. Dans ce sens l'algorithme de la sécante est une approximation de l'algorithme de Newton.

Algorithme 3 (Algorithme de la sécante).

Entrée : Deux approximations initiale x_0 et x_1,
la précision ϵ souhaitée et **N** le nombre maximum d'itérations
Sortie : Une approximation c de la racine x^* de f ou bien un message d'erreur.

1. $n := 2$;
   $q_0 := f(x_0)$;
   $q_1 := f(x_1)$;
2. **Tant que** $n \leq N + 1$ **Faire**
   $c := x_1 - q_1 \frac{x_1-x_0}{q_1-q_0}$;
3. **Si** $\mid c - x_1 \mid \leq \epsilon$ **Alors** Imprimer(c) ; **Fin.**
4. $n := n + 1$;
   $x_0 := x_1$; $x_1 := c$;
   $q_0 := q_1$; $q_1 := f(c)$;
   Fin Tant que ;
   Imprimer(*La méthode a échoué après N itérations*) ;

Fin.

2.1.4 Méthode du point fixe

On peut toujours transformer un problème du type $f(x) = 0$ en un problème de la forme $g(x) = x$ et ce d'une infinité de façons. Par exemple, la méthode de Lagrange est donnée par $x_{n+1} = g(x_n)$ où $g(x) = \frac{af(x)-xf(a)}{f(x)-f(a)}$ (on pose alors $x_0 = b$). Il faut toutefois noter que de telles transformations introduisent parfois des solutions parasites. Par exemple, on peut écrire $f(x) = \frac{1}{x} - 3 = 0$ ou encore $x = 2x - 3x^2 = g(x)$ et 0 est solution de la seconde équation mais pas de la première.

Le théorème suivant caractérise les fonctions g qui donnent des suites convergentes :

Théorème 2.1.3. *Si dans l'intervalle $[a, b]$, g vérifie les conditions suivantes :*

1. *$x \in [a, b]$ alors $g(x) \in [a, b]$,*
2. *l'application g est strictement contractante, c'est-à-dire $max_{x\in[a,b]} \mid g'(x) \mid = L < 1$.*

Alors pour tout $x_0 \in [a, b]$, la suite définie par $x_{n+1} = g(x_n)$ converge vers l'unique point fixe x^* de g sur $[a, b]$.

Il est souvent délicat de déterminer un intervalle $[a, b]$ dans lequel les hypothèses du théorème 2.1.3 sont vérifiées. Si on peut estimer $\mid g'(x^*) \mid$, on a le résultat suivant :

Théorème 2.1.4. *Soit x^* une solution de l'équation $x^* = g(x^*)$. Si $g^{'}$ est continue et $\mid g^{'}(x^*) \mid < 1$ alors il existe un intervalle $[a, b]$ contenant x^* pour lequel la suite définie par $x_0 \in [a, b]$ et $x_{n+1} = g(x_n)$ converge vers x^*.*

Proposition 2.1.5. *Soit x^* une solution de l'équation $x^* = g(x^*)$. Si $g^{'}$ est continue au voisinage de x^* et si $\mid g^{'}(x^*) \mid > 1$ alors pour tout $x_0 \in [a, b]$ différent de x^*, la suite $x_{n+1} = g(x_n)$ ne converge pas vers x^*.*

2.2 Méthodes d'accélération de la convergence

On veut savoir si la suite définie par $x_{n+1} = g(x_n)$ converge rapidement et comment se comporte l'erreur $e_n = x_n - x^*$ d'une itération à la suivante ?

2.2.1 Procédé Δ^2 d'Aitken

Dans les méthodes d'ordre 1 (i.e. $g'(x^* \neq 0)$), on considère la suite

$$y_n = x_n - \frac{(x_{n+1} - x_n)^2}{x_{n+2} - 2x_{n+1} + x_n} \quad .$$

C'est une meilleure approximation de x^* que x_n. Lorsque x_n est proche de la solution x^*, $(x_{n+1} - x_n)^2$ et $x_{n+2} - 2x_{n+1} + x_n$ sont voisins de 0. Le calcul de y_n est alors instable. On utilise alors une autre écriture équivalente à la précédente mais plus stable :

$$y_n = x_{n+1} + \frac{1}{\frac{1}{x_{n+2} - x_{n+1}} - \frac{1}{x_{n+1} - x_n}} \quad .$$

2.2.2 Méthode de Steffensen

On poursuit les itérations en remplaçant x_n par y_n et on calcule ainsi $g(y_n) = z_n$ et $g(g(y_n)) = g(z_n)$. On construit formellement la méthode de Steffensen $z_{n+1} = G(z_n)$ où $G(x) = \frac{xg(g(x)) - g(x)^2}{g(g(x)) - 2g(x) + x}$. La méthode de Steffensen est d'ordre au moins 2 si (x_n) est d'ordre 1 et si (x_n) est d'ordre p alors (z_n) est d'ordre $2p - 1$.

2.3 Exercices d'Applications

1. Soit $f(x) = x^2 - \exp(-1 - x^2)$ une fonction définie sur $[0, 5]$.
   (a) Tracer f, f' et f'' sur $[0, 5]$.
   (b) Montrer qu'il existe une racine unique de f sur $[0, 5]$.
   (c) Appliquer la méthode de Newton avec $x_0 = 5$.
   (d) Est-ce que la méthode de Newton converge pour $x_0 = 0$? justifier votre réponse.
   (e) Appliquer la méthode de la sécante à f avec $x_0 = 0$. Que remarquez-vous ?
   (f) Trouver une fonction g vérifiant $g(x) = x$ si et seulement si $f(x) = 0$ qui assure les conditions du théorème du point fixe. Appliquer la méthode des itérations successives.
   (g) Citer deux procédés pour accélérer la convergence des algorithmes.
2. Refaire l'exercice précédent avec $f(x) = x^3 - 4x + 1$ sur $[0, 0.5]$ et $g(x) = x - f(x)$.
3. (a) Donner le théorème de convergence globale de la méthode de Newton pour une fonction de classe C^2.
   (b) Ecrire l'algorithme de Newton qui prend en entrée les points x_0, $a < b$ et une fonction f et rend la racine de $f(x) = 0$ sur $[a, b]$ ou bien un message d'erreur.
   (c) Calculer la racine de $f(x) = -2x^3 + x^2 - 2x + 2$ sur l'intervalle $[0.5, 1]$ avec $x_0 = 1$. La précision des calculs est à 10^{-6} près.
4. Soit f une fonction de $\mathbf{R}$ dans $\mathbf{R}$ telle que

$$f(x) = x^3 + 3x^2 - 1 \qquad \textit{pour} \qquad x \in \mathbf{R}.$$

   (a) Montrer que l'équation $f(x) = 0$ admet une unique solution sur $[0, 1]$.
   (b) Nous désirons déterminer la solution de l'équation $f(x) = 0$ avec la méthode de Newton sur $[0, 1]$. En quoi consiste cette méthode ?
   (c) Ecrire l'algorithme général de la méthode de Newton qui prend en entrée les points x_0, $a < b$ et une fonction f et rend la racine de $f(x) = 0$ sur $[a, b]$ ou bien un message d'erreur.
   (d) Est-ce que la fonction f vérifie les conditions du théorème de convergence globale de la méthode de Newton pour $x_0 = 1$. Donner le théorème.
   (e) Calculer la racine de f sur l'intervalle $[0, 1]$ avec une erreur de 10^{-6}.
5. On cherche le zéro réel de $f(x) = x^3 + 4x^2 - 10$.
   (a) En choisissant $x_0 = 1.5$, appliquer la méthode du point fixe aux transformations suivantes :
      i. $g_1(x) = x - x^3 - 4x^2 + 10$
      ii. $g_2(x) = \frac{1}{2}\sqrt{10 - x^3}$
      iii. $g_3(x) = \sqrt{\frac{10}{4+x}}$

      On remarquera que l'algorithme ne converge pas pour g_1, alors qu'il converge mais à des vitesses différentes pour les autres.
   (b) Approximer la racine de f avec la méthode de Newton et donner un nombre d'itérations n qui donne une précision de x^* à 10^{-9} près.
6. Implémenter l'ensemble des algorithmes de ce chapitre sur Mupad.

Chapitre 3

Analyse Numérique matricielle

L'analyse numérique matricielle s'intéresse principalement à deux types de problèmes :
– Résoudre un système linéaire $Ax = b$;
– Calculer les valeurs propres et vecteurs propres d'une matrice donnée.
Nous étudierons pour cela des algorithmes destinés à des matrices de grande taille. Comme il s'agit de calculs volumineux sur machine, à chaque opération des erreurs systématiques dites d'arrondi son générées. Il convient donc de se préoccuper de l'influence de ces petites erreurs sur la solution du problème et donc d'étudier la stabilité. Nous nous intéresserons d'autre part, au coût de ces algorithmes en termes d'opérations élémentaires.

3.1 Rappels et compéments sur les matrices

Les matrices diagonales et triangulaires jouent un rôle particulier en analyse numérique matricielle pour la raison évidente que les systèmes linéaires avec de telles matrices sont de résolution très simple. Aussi nous nous intéresserons ici aux changement de base permettant d'accéder à des structures de ce type.

Nous considérons un espace vectoriel de dimension n sur $\mathbf{R}$ ou sur $\mathbf{C}$. Toutes les matrices considérées dans la suite sont des matrices carrées.

Rappelons qu'une matrice est dite *régulière* si elle est inversible, sinon elle est *singulière*. Une matrice A est dite *normale* si $AA^* = A^*A$ où A^* est la matrice adjointe définie par $< Au, v >=< u, A^*v >$ pour tout $u, v \in \mathbf{C}^n$. Le signe $<,>$ dénote le produit scalaire dans $\mathbf{C}^n$, soit $< u, v >= {}^t\bar{u}v$. Une matrice *unitaire* est une matrice vérifiant $A^*A = AA^= I_n$. Elle est *hermitienne* si $A^* = A$.
Une matrice est dite symétrique si elle est réelle et si ${}^tA = A$ (dans ce cas, ${}^tA = A^*$). La matrice A est orthogonale si ${}^tAA = A\,{}^tA = I_n$.

Théorème 3.1.1. *Etant donné une matrice carrée A, il existe une matrice unitaire U telle que $U^{-1}AU$ soit triangulaire. Etant donné une matrice normale A, il existe une matrice unitaire U telle que $U^{-1}AU$ soit diagonale.*

3.2 Normes matricielles

Soit V un espace vectoriel de dimension n sur $\mathbf{R}$ ou $\mathbf{C}$, on peut définir plusieurs normes équivalentes sur V dont :

$$\begin{array}{lcll} \| x \|_1 & = & \sum_{i=1}^{i=n} |x_i|, & x = {}^t(x_1, x_2, \ldots, x_n) \in V \\ \| x \|_\infty & = & \mathrm{max}_{i=1..n}|x_i| & \\ \| x \|_2 & = & \sqrt{\sum_{i=1}^{i=n} |x_i|^2} = \sqrt{{}^t\bar{x}x}, & \text{norme de schur} \\ \| x \|_p & = & (\sum_{i=1}^{i=n} |x_i|^p)^{\frac{1}{p}}. & \end{array}$$

Si V est muni d'une norme $\| \, . \, \|$, on définit sur l'ensemble des matrices carrées d'ordre n la norme suivante :

$$\| A \| = sup_{x \neq 0} \frac{\| Ax \|}{\| x \|} = max_{\|x\|=1} \| Ax \|$$

Cette quantité définit bien une norme puisqu'elle vérifie

$$\begin{array}{lcl} \| A \| = 0 & \Rightarrow & A = 0 \\ \| \lambda A \| & = & |\lambda| \, \| A \| \\ \| A + B \| & \leq & \| A \| \, \| B \| \end{array}$$

3.2.1 La fatorisation de Cholesky

3.2.2 La factorisation LU d'une matrice

3.2.3 La factorisation QR

3.2.4 Calcul de l'inverse d'une matrice

3.3 Méthodes itératives

3.3.1 Introduction

3.3.2 La méthode de Jacobi

3.3.3 La méthode de Gauss-Seidel

3.3.4 Vitesse de convergence d'une méthode itérative

3.4 Valeurs propres et vecteurs propres

3.4.1 La méthode de la puissance

3.4.2 La méthode de Givens-Householder (bissection)

3.4.3 Conditionnement d'un problème de valeurs propres

Annexe A

Exercices d'Applications

1. (a) Trouver le polyôme P du secons degré passant par $(0,0)$, $(1,1)$, $(2,8)$.
   (b) La fonction x^3 prend les valeurs de P. Utiliser la formule vue en cours pour exprimer l'erreur d'interpolation commise en remplaçant x^3 par le polynôme d'interpolation P.
2. (a) Interpoler e^x aux points 0, $\frac{1}{2}$ et 1 par un polynôme du second degré.
   (b) Trouver une expression pour l'erreur d'interpolation et une borne de l'erreur indépendante de ζ.
3. (a) Sachant que la température initiale d'un objet est de $0^{\mathbf{o}}\mathbf{C}$, puis de $5^{\mathbf{o}}\mathbf{C}$ après une minute et $3^{\mathbf{o}}\mathbf{C}$ après deux minutes, quelle a été la température maximum et quant a-t-elle été atteinte ?
   (b) Quelle est la température moyenne entre 1 et 2 minutes ?
4. (a) Calculer $\int_0^{\frac{\pi}{2}} sin^2(x)\, dx$ en utilisant la formule du trapèze et de Simpson.
   (b) Comparer avec le résultat exact.
5. (a) Donner une approximation de l'EDO $y'(x) = -10y(x)$ pour $x \in [0,2]$ et $y(0) = 1$ en utilisant la méhode d'Euler avec $h = 1$.
   (b) Donner une interpolation polynomiale de la solution y.
   (c) La solution exacte étant e^{-10x}, la comparer avec l'approximation de la solution.