الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة محمد الصديق بن يحي جيجل
Université Mohamed Essedik Ben Yahia de Jijel

Cours

De Méthodes Avancées D'Analyse Numérique

Destiné

Aux Etudiants inscrits en Master 1

Spécialité:

Electronique et Systèmes de Communication

Préparé par:

Dr.Tahar BRAHIMI

Préface

L'analyse numérique est une discipline des mathématiques. Elle s'intéresse tant aux fondements théoriques qu'à la mise en pratique des méthodes permettant de résoudre, par des calculs purement numériques, des problèmes d'analyse mathématique.

Les méthodes numériques trouvent des applications dans de nombreux domaines : la physique, les sciences biologiques, les sciences de l'ingénieur, l'économie et la finance.

Un des objectifs de ce support de cours est d'exposer les fondements mathématiques de différentes méthodes numériques en analysant leurs propriétés théoriques et en présentant leurs avantages et inconvénients à l'aide d'exemples illustratifs.

Ce support de cours, intitulé « Méthodes Avancées d'Analyse Numérique», est destiné principalement aux étudiants inscrits en Master 1, Spécialité : Electronique et Systèmes de Communication. Ce cours est divisé en six chapitres qui sont les suivants:

Le chapitre 1 traite du problème de résolution d'équations non linéaires. Il comporte deux parties. La première partie présente les différentes méthodes pour la résolution d'équations non linéaires (Bissection, Points fixes, Newton-Raphson, et la Sécante). La seconde partie, quant à elle, se consacre principalement pour la résolution des systèmes d'équations non linéaires, en exposant les méthodes les plus usuelles d'évaluation utilisées, en l'occurrence les méthodes des Point fixes et la méthode de Newton en dimension n.

Le chapitre 2 aborde les principales méthodes (directes et indirectes) de résolution des systèmes d'équations linéaires, à savoir les méthodes de (Cramer, Gauss, Gauss-Jordan, Jacobi, Gauss-Seidel), la décomposition LU (décomposition de Crout et méthode de Cholesky), la décomposition QR et la décomposition SVD.

Le chapitre 3 porte sur la dérivation et l'intégration numérique. Il aborde le problème qui consiste à obtenir des approximations des différentes dérivées et intégrales d'une fonction f(x) connue seulement en quelques points. Le chapitre présente d'abord différentes formules de différences finies d'ordre 1 et 2 pour approximer les dérivées d'une fonction. Il se termine par illustrer les méthodes couramment utilisée pour l'intégration numérique, à savoir les Méthode des trapèzes composée, et Méthode de Simpson 1/3 composée en offrant des exemples illustratifs pour permettre une meilleure compréhension.

Le chapitre 4 s'intéresse à la résolution numérique d'équations différentielles qui est probablement le domaine de l'analyse numérique où les applications sont les plus nombreuses. Le chapitre passe en revue diverses méthodes de résolution: méthode d'Euler,

méthode de Taylor, méthodes de Rung-Kutta d'ordre 2 (méthode d'Euler modifiée, méthode du point Milieu), et des méthodes plus complexes telle que la méthode de Rung-Kutta d'ordre 4 qui permet d'obtenir des résultats avec une grande précision. Ceci a été illustré par un tableau comparatif inclut qui évalue la performance de trois méthodes différentes de résolution numérique.

Le chapitre 5 s'intitule interpolation polynomiale. Il est très étroitement relié au chapitre 3 puisqu'ils tendent à répondre à diverses facettes d'un même problème. Il s'agit de construire une approximation d'une fonction f(x) connue seulement en (n+1) points (points de collocation) pour tout x. Ce chapitre décrit quatre méthodes d'interpolation, à savoir matrice de Vandermonde, Interpolation de Lagrange, Polynôme de Newton, et la plus performante: Splines cubiques.

Le dernier chapitre (chapitre 6) traite du problème d'approximation au sens des moindres carrées. Il fait appel aux méthodes : Droite des moindres carrées et Parabole des moindres carrées. Une démonstration des deux théorèmes énoncés dans ce chapitre est également effectuée.

Sommaire

Chapitre 1: Résolution d'équations et des systèmes d'équations non linéaires

I. Résolution d'équations non linéaires

I.1 Introduction:

Le numéricien est souvent confronté à la résolution d'équations algébriques de la forme: $f(x)=0$, et ce, dans toutes sortes de contextes. Introduisons la terminologie suivante:

Définition:

Une valeur de x solution de $f(x)=0$ est appelée une racine ou zéro de la fonction $f(x)$. Elle est notée r.

Il est possible d'obtenir une formule générale pour un polynôme du $4^{\text{ième}}$ degré, et encore, beaucoup plus complexe pour celui de $3^{\text{ième}}$ ordre. Par contre, il n'existe pas de formule permettant de trouver les racines des polynômes de degré plus grand ou égal à 5. Il faudra donc recourir aux méthodes numériques. Dans ce qui suit, nous présenterons plusieurs techniques de résolution. Les algorithmes de recherche d'une racine reposent plus ou moins directement sur le théorème suivant:

Théorème des valeurs intermédiaires:

Soit f une fonction continue dans $[a,b]$; alors pour tout réel F compris entre $f(a)$ et $f(b)$, il existe au moins un réel C de $[a,b]$ tel que $f(c)=F$. En d'autres termes, si $f(a)$ et $f(b)$ sont de signes contraires ($f(a)\times f(b)<0$), la fonction continue f présente au moins un zéro (une racine) entre a et b. ($\exists$ au moins $r\in[a,b]$ tel que $f(r)=0$).

I.2 Méthode de Bissection (ou de Dichotomie):

Soit $f(x)$ une fonction non linéaire dérivable et continue sur l'intervalle $[a,b]$.

L'idée: construire une suite d'intervalles de plus en plus petits contenant une racine isolée de l'équation: $f(x)=0$. Soit une fonction $f(x)$ continue qui change de signe et passe du positif au négatif ou vice versa. Soit $[a,b]$ un intervalle ayant un changement de signe: $f(a)\times f(b)<0$

On pose $m = \frac{a+b}{2}$ qui est le point milieu de l'intervalle. Il suffit de déterminer entre les intervalles $[a,m]$ et $[m,b]$ celui qui possède encore un changement de signe. La racine se trouvera forcément dans cet intervalle. Cela nous amène à l'algorithme suivant:

Algorithme de la bissection:

1. Etant donné un intervalle $[a,b]$ pour lequel $f(x)$ possède un changement de signe.
2. Etant donné ε, un critère d'arrêt, et N, le nombre maximal d'itérations.
3. Poser $m = \frac{a+b}{2}$
4. Si $m = \frac{|b-a|}{2|m|} < \varepsilon$

   Convergence atteinte

   Ecrire la racine m

   Ecrire $f(x)$

   Arrêt
5. écrire $a, b, m, f(a), f(b), f(m)$
6. Si $f(a) \times f(m) < 0$, alors $b = m$
7. Si $f(m) \times f(b) < 0$, alors $a = m$
8. Si le nombre maximal d'itérations est atteint:

   Convergence atteinte en N itérations.

   Arrêt
9. Retour à l'étape 3.

Remarques:

Dans l'algorithme précédent, il faut prendre garde au cas où la racine recherchée est 0, il y a un risque de division par 0 au cours de l'évaluation de l'erreur relative.

Il est parfois utile d'introduire un critère d'arrêt sur la valeur de $f(x)$ qui doit tendre également vers 0.

La longueur de l'intervalle entourant la racine est divisée par 2 à chaque itération. Ainsi, ce constat permet de déterminer à l'avance le nombre d'itérations nécessaires pour obtenir une certaine erreur absolue Δr

Soit $L = b - a$: longueur de l'intervalle de départ.

Après une itération, le nouvel intervalle est de longueur $\frac{L}{2}$

Après n itérations, le nouvel intervalle est de longueur $\frac{L}{2^n}$

Si l'on veut connaitre la valeur de n nécessaire pour avoir $\frac{L}{2^n} < \Delta r$ il suffit de résoudre cette équation en fonction de n et l'on trouve $n > \frac{\ln(\frac{L}{\Delta r})}{\ln 2}$

Exemple: soit $f(x) = x^3 + x^2 - 3x - 3$, $b = 2$; $a = 1$; $L = b - a = 2 - 1 = 1$

Si l'on veut une erreur absolue plus petite que 0.5×10^{-2}

$$\frac{\ln(\frac{1}{0.5 \times 10^{-2}})}{\ln 2} = 7.64 \text{ itérations } \Rightarrow \text{n=8}$$

On fera donc 8 itérations pour s'assurer de cette précision.

I.3 Méthode des points fixes:

Définition: Un point fixe de la fonction $f(x)$ est toute solution de $x = g(x)$. Un point fixe n'est pas une racine de l'équation $0 = g(x)$. Il est une solution de l'équation $x = g(x)$. Géométriquement, les points fixes de la fonction $g(x)$ sont les points d'intersection entre les courbes $y = g(x)$ et la courbe $y = x$. (voir Figure 1.1).

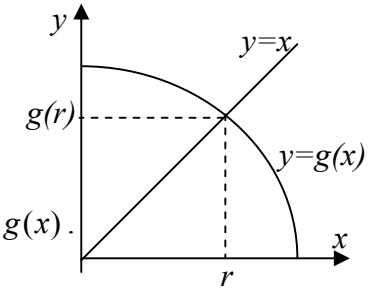

Figure 1.1

L'algorithme $\begin{cases} x_0 \\ x_{n+1} = g(x_n) \end{cases}$

Permet de déterminer les points fixes (itérations des points fixes) de $g(x)$.

L'algorithme le plus complet est le suivant:

Algorithme des points fixes:

Entrées: terme initial x_0

Fonction $g(x)$ (déduite de $f(x)$)

Condition d'arrêt

Initialisation de la condition d'arrêt

Tant que la condition d'arrêt n'est pas vérifiée faire

$x_{n+1} = g(x_n)$

Mise à jour de la condition d'arrêt

Fin Tant que

Retourner x

Convergence de la méthode des points fixes:

Théorème:

Soit r, une valeur qui est à la fois une racine de $f(x)$ et un point fixe de $g(x)$: $f(r)=0; r=g(r)$. Soit $g(x)$ une fonction continue dans l'intervalle $I=[a,b]$ et telle que $g(x)\in I$ pour tout x dans I. Si de plus $g'(x)$ existe et si $|g'(x)|\le k<1$ pour tout $x\in I$, alors les points x_0 de I convergent vers l'unique point fixe r de I.

Définition:

Le taux de convergence d'une méthode des points fixes est donné par $|g'(r)|$

Un point fixe r de la fonction g(x) est dit:

Attractif si : $|g'(r)|<1$

Répulsif si : $|g'(r)|>1$

Le cas $|g'(r)|=1$ est indéterminé.

Remarque: il est possible que la méthode des points fixes converge dans le cas où : $|g'(r)|=1$. La convergence dans ce cas est au mieux extrêmement lente car $|g'(r)|$ est prés de 1.

Exemple:

Soit $f(x)=x^2-2x-3$. On a $r_1=3,\ r_2=-1$. Il ya une infinité de façons différentes pour transformer cette équation sous la forme $x=g(x)$. On choisi trois au hasard

$x=\sqrt{2x+3}=g_1(x)$ (en isolant x^2)

$x=\dfrac{3}{x-2}=g_2(x)$ (en écrivant $x(x-2)-3=0$)

$x=\dfrac{x^2-3}{2}=g_3(x)$ (en isolant x de -2 x)

Si l'on applique la méthode des points fixes en partant de $x_0=4,$ on obtient:

Pour $g_1(x)$:
$$\begin{aligned} x_1 &= g_1(4)=3.3166248\\ x_2 &= g_1(3.3166248)=3.1037477\\ x_3 &= g_1(3.1037477)=3.0343855\\ &\vdots\\ x_{10} &= g_1(3.0000470)=3.0000157 \end{aligned}$$

On a convergence vers $r1$.

Pour $g_2(x)$:
$$\begin{aligned} x_1 &= g_2(4) = 1.5 \\ x_2 &= g_2(1.5) = -6 \\ x_3 &= g_2(-6) = -0.775 \\ &\vdots \\ x_{10} &= g_2(-0.9989841) = -1.0003387 \end{aligned}$$

Les itérations convergent vers $r2 = -1$.

Pour $g_3(x)$:
$$\begin{aligned} x_1 &= g_3(4) = 6.5 \\ x_2 &= g_3(6.5) = 19.625 \\ x_3 &= g_3(-6) = 191.0703 \\ x_4 &= g_3(191.0703) = 18252.43 \end{aligned}$$

Visiblement, les itérations tendent vers l'infini et aucune des 2 solutions possibles ne sera atteinte.

On a: $g_1'(x) = \dfrac{1}{\sqrt{2x+3}}$; $g_2'(x) = \dfrac{-3}{(x-2)^2}$; $g_3'(x) = x$

Taux de convergence		
	$r_1 = 3$	$r_2 = -1$
$g_1'(r)$	0.3333	1
$g_2'(r)$	-3	-0.3333
$g_3'(r)$	3	-1

Ce tableau aide à comprendre les résultats obtenus précédemment.

$\left|g_1'(3)\right| < 1$: Convergence de $g_1(x)$ vers 3.

$\left|g_2'(-1)\right| < 1$: Convergence de $g_2(x)$ vers -1.

I.4 Méthode de Newton-Raphson:

Notons par r une racine (exacte) recherchée et par x_0 une valeur approchée de r. Le développement de Taylor d'ordre 2 de $f(x)$ nous donne:

$$f(r) = f(x_0) + f'(x_0)(r - x_0) + \frac{f''(\varepsilon)}{2}(r-x_0)^2 \qquad \text{où} \quad \varepsilon \in (r, x_0)$$

Et comme $f(r) = 0$, en supposant $f'(x) \neq 0$, on aura

$$r = \underbrace{x_0 - \frac{f(x_0)}{f'(x_0)}}_{x_1} - \frac{f''(\varepsilon)}{2f(x_0)}(r-x_0)^2 \searrow 0$$

En itérant le procédé on trouve la formule :

$x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}$ $\quad n = 0,1,2$ qui représente la formule de récurrence de Newton.

Géométriquement:

On a: $tg(\alpha) = \dfrac{f(x_n)}{\Delta x_n} = f'(x_n)$ (voir figure 1.2)

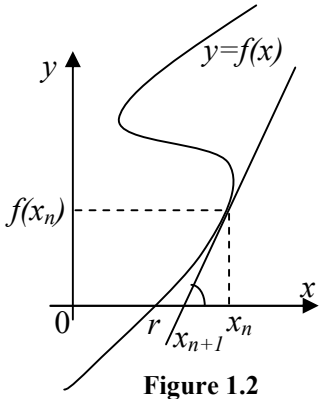

Figure 1.2

Où $\Delta x_n = x_n - x_{n+1}$

Et donc

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

Algorithme de Newton:

Entrées: terme initial x_0

Fonctions $f(x), f'(x)$, tolérance.

Condition d'arrêt (ε (tolérance), nombre maximal d'itérations)

Initialisation de la condition d'arrêt

Tant que la condition d'arrêt n'est pas vérifiée faire

$x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}$;

Mise à jour de la condition d'arrêt

Fin Tant que

Retourner x

Remarque:

L'algorithme de la méthode de Newton est un cas particulier de celui de la méthode des points fixes où $g(x) = x - \dfrac{f(x)}{f'(x)}$

Conditions de convergence:

Soit $f(x) = 0$, si $f \in C^2[a,b]$ (deux fois dérivable et $f''(x)$ continue) vérifie :

* $f(a) \times f(b) < 0$

* $\forall x \in [a,b],\ f'(x) \neq 0$ et $f''(x) \neq 0$

* $\exists\, x_0 \in [a,b]$ tant que $f(x_0) \times f''(x_0) > 0$

I.5 Méthode de la Sécante:

La méthode de Newton possède de grands avantages, mais elle nécessite le calcul de la dérivée de $f(x)$. Dans certaines circonstances, par exemple si $f(x)$ est complexe, sa dérivée peut être difficile à évaluer et peut résulter une expression complexe (où si $f(x)$ provient d'un calcul expérimental par exemple), on ne peut calculer $f'(x)$. On contourne ces difficultés en remplaçant $f'(x)$ par l'expression suivante:

$$f'(x_n) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$$

Algorithme de la méthode de la Sécante:

Entrées : x_0, x_1 : 2 valeurs initiales de la solution

Fonctions $f(x)$, $f'(x)$, tolérance

Condition d'arrêt

Initialisation de la condition d'arrêt

Tant que la condition d'arrêt n'est pas vérifiée faire

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

Mise à jour de la condition d'arrêt

Fin Tant que

Retourner x

II. Résolution des systèmes d'équations non linéaires

II.1 Introduction:

Le problème consiste à trouver le ou les vecteurs $\vec{X} = \left[x_1\, x_2\, x_3 \cdots x_n\right]^T$ vérifiant les n équations non linéaires suivantes:

$$\begin{aligned} f_1(x_1, x_2, x_3, \cdots x_n) &= 0 \\ f_2(x_1, x_2, x_3, \cdots x_n) &= 0 \\ f_3(x_1, x_2, x_3, \cdots x_n) &= 0 \\ &\vdots \\ f_n(x_1, x_2, x_3, \cdots x_n) &= 0 \end{aligned}$$

Où les f_i sont des fonctions de n variables que nous supposons différentiables. La méthode la plus importante et la plus utilisée est la méthode de Newton.

II.2 Méthode de Newton-Raphson en dimension *n*:

Supposons qu'on cherche la solution du système d'équations non linéaires suivant:

$$\begin{cases} f_1(x,y)=0 \\ f_2(x,y)=0 \end{cases} \quad (*)$$

Les équations $f_1(x,y)=0$ et $f_2(x,y)=0$ définissent implicitement les courbes dans le plan xy. Ainsi la solution du système (*) est un point (p,q) où les deux courbes se croisent. (i.e: s $f_1(p,q)=0$ et $f_2(p,q)=0$)

Pour résoudre une équation non linéaire, la méthode de Newton fait appel à l'équation suivante : $x_{k+1}=x_k-\dfrac{f(x_k)}{f'(x_k)}$, qui peut se mettre sous la forme:

$$x_{k+1}-x_k=-\frac{f(x_k)}{f'(x_k)}$$

$$f'(x_k)(x_{k+1}-x_k)=-f(x_k)$$

La méthode de Newton en dimension n repose sur une formule analogue à l'équation précédente:

$$J(\vec{X}^k)(\vec{X}^{k+1}-\vec{X}^k)=-\vec{F}(\vec{X}^k) \quad (a)$$

$$\text{Avec } \delta\vec{X}=(\vec{X}^{k+1}-\vec{X}^k)$$

Où J est la matrice jacobienne formée des dérivées partielles de $\vec{F}$ par rapport à $\vec{X}$

$$J(\vec{X}^k)=\begin{pmatrix} \dfrac{\partial f_1(\vec{X}^k)}{\partial x_1} & \dfrac{\partial f_1(\vec{X}^k)}{\partial x_2} & \cdots & \dfrac{\partial f_1(\vec{X}^k)}{\partial x_n} \\ \dfrac{\partial f_2(\vec{X}^k)}{\partial x_1} & \dfrac{\partial f_2(\vec{X}^k)}{\partial x_2} & \cdots & \dfrac{\partial f_2(\vec{X}^k)}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \dfrac{\partial f_n(\vec{X}^k)}{\partial x_1} & \dfrac{\partial f_n(\vec{X}^k)}{\partial x_2} & \cdots & \dfrac{\partial f_n(\vec{X}^k)}{\partial x_n} \end{pmatrix} \quad (b)$$

De (a) on a :

$$\delta\vec{X}=-J(\vec{X}^k)^{-1}\vec{F}(\vec{X}^k)$$

$$\Rightarrow \vec{X}^{k+1}=\vec{X}^k+\delta\vec{X} \quad (c)$$

$$\vec{F}(\vec{X}^k)=\begin{bmatrix} f_1(\vec{X}^k) \\ f_2(\vec{X}^k) \\ \vdots \\ f_n(\vec{X}^k) \end{bmatrix}, \quad \delta\vec{X}=\begin{bmatrix} \delta x_1 \\ \delta x_2 \\ \vdots \\ \delta x_n \end{bmatrix}$$

Pour le cas de 2 équations, (c) devient :

$$\begin{cases} x_1^1 = x_1^0 + \delta x_1 \\ x_2^1 = x_2^0 + \delta x_2 \end{cases}$$

Il est à noter que $\vec{X}^{k+1}$, $\vec{X}^k$, et $\vec{F}(\vec{X}^k)$ sont des vecteurs et que $J(\vec{X}^k)$ est une matrice. La méthode de Newton fonctionne si J est inversible puisque la formule utilise l'inverse de J.

Algorithme de la méthode de Newton en dimension *n*:

1. Etant donné ε un critère d'arrêt.
2. Etant donné N, le nombre maximal d'itération.
3. Etant donné $\vec{X}_0^0=\left[x_1^0\ x_2^0\ x_3^0 \cdots x_n^0\right]^T$ une approximation initiale de la solution du système.
4. Evaluer la fonction $\vec{F}(\vec{X}^0)=\begin{bmatrix} f_1(\vec{X}^0) \\ f_2(\vec{X}^0) \\ \vdots \\ f_n(\vec{X}^0) \end{bmatrix}=\begin{bmatrix} f_1(x_1^0, x_2^0, x_3^0, \cdots x_n^0) \\ f_2(x_1^0, x_2^0, x_3^0, \cdots x_n^0) \\ \vdots \\ f_n(x_1^0, x_2^0, x_3^0, \cdots x_n^0) \end{bmatrix}$
5. Evaluer la matrice jacobienne (équation (b))
6. Résoudre le système linéaire : $J(\vec{X}^k)\delta\vec{X}=-\vec{F}(\vec{X}^k)$ pour $\delta\vec{X}$ (D)
7. Calculer le point suivant (équation (c)) : $\vec{X}^{k+1}=\vec{X}^k+\delta\vec{X}$
8. Si $\left\|\frac{\delta\vec{X}}{\vec{X}^{k+1}}\right\|<\varepsilon$ | $\left\|\vec{F}(\vec{X}^{(k)})\right\|\le\varepsilon$

   Convergence atteinte

   Ecrire la solution

   Arrêt
9. Si le nombre maximal N d'itération est atteint :

   Convergence non atteinte en N itérations.

   Arrêt
10. Retour à l'étape 5

Exemple: on cherche à trouver l'intersection de la courbe $x_2 = e^{x_1}$ et du cercle de rayon 4 centré à 0 d'équation $x_1^2 + x_2^2 = 16$. L'intersection de ces courbes est une solution de:

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

Traçage des courbes:

```
ezplot('exp(x1)-x2')
hold on
ezplot('x1^2+x2^2-16')
grid
axis([-4 4 -4 4])
```

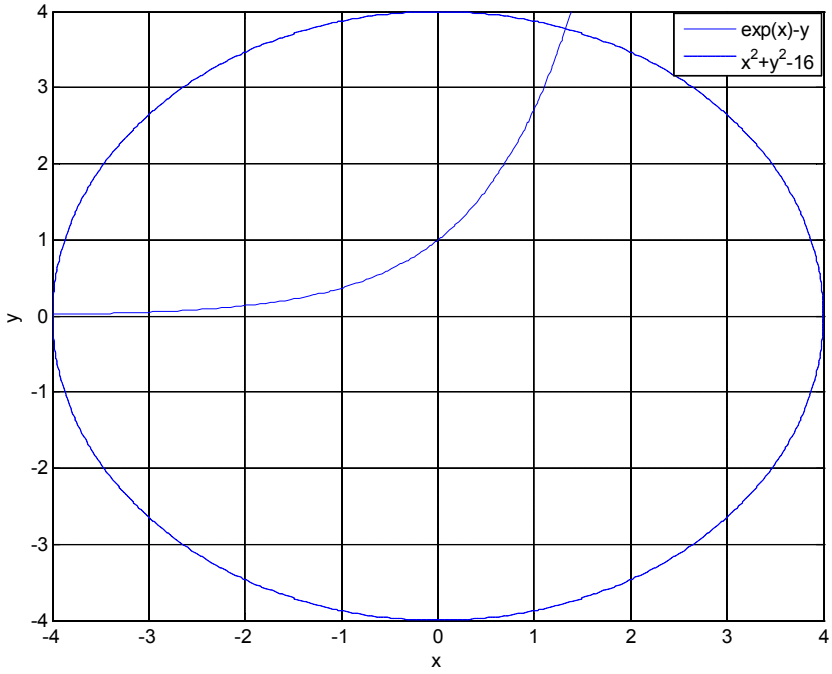

Figure 1.3

Calcul du J :

$$J(x_1, x_2) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{pmatrix} = \begin{pmatrix} e^{x_1} & -1 \\ 2x_1 & 2x_2 \end{pmatrix}$$

D'après la figure 3:

La première solution est proche de (-4,0).

La deuxième solution est proche de (2.8,2.8).

Soit $\vec{X}^0 = \begin{bmatrix} 2.8 & 2.8 \end{bmatrix}^T$

Itération 1 :

$$\begin{pmatrix} e^{2.8} & -1 \\ 2(2.8) & 2(2.8) \end{pmatrix} \begin{pmatrix} \delta x_1 \\ \delta x_2 \end{pmatrix} = -\begin{pmatrix} e^{2.8} - 2.8 \\ (2.8)^2 + (2.8)^2 - 16 \end{pmatrix}$$

C'est-à-dire :

$$\begin{pmatrix} 16.445 & -1 \\ 5.6 & 5.6 \end{pmatrix} \begin{pmatrix} \delta x_1 \\ \delta x_2 \end{pmatrix} = -\begin{pmatrix} 13.645 \\ -0.380 \end{pmatrix}$$

Dont la solution est : $\delta\vec{X} = \begin{bmatrix} -0.77890 & 0.83604 \end{bmatrix}^T$

$$\begin{cases} x_1^1 = x_1^0 + \delta x_1 = 2.8 - 0.77890 = 2.0211 \\ x_2^1 = x_2^0 + \delta x_2 = 2.8 + 0.83604 = 3.63604 \end{cases}$$

Itération 2 : à partir de $\left[2.0211 \quad 3.63604\right]^T$

Le système (D) devient :

$$\begin{pmatrix} e^{2.0211} & -1 \\ 2(2.0211) & 2(2.0211) \end{pmatrix}\begin{pmatrix} \delta x_1 \\ \delta x_2 \end{pmatrix} = -\begin{pmatrix} e^{2.0211} - 3.63604 \\ (2.0211)^2 + (3.63604)^2 - 16 \end{pmatrix}$$

C'est-à-dire :

$$\begin{pmatrix} 7.5466 & -1 \\ 4.0422 & 7.2721 \end{pmatrix}\begin{pmatrix} \delta x_1 \\ \delta x_2 \end{pmatrix} = -\begin{pmatrix} 3.9106 \\ 1.3056 \end{pmatrix}$$

Dont la solution est : $\delta\vec{X} = \left[-0.5048 \quad 0.10106\right]^T$

On a maintenant :

$$\begin{cases} x_1^2 = x_1^1 + \delta x_1 = 2.0211 - 0.50480 = 1.5163 \\ x_2^2 = x_2^1 + \delta x_2 = 3.63604 + 0.10106 = 3.7371 \\ \vdots \end{cases}$$

Itération 5 : à partir de $\left[1.3280 \quad 3.7731\right]^T$

On doit résoudre: $\begin{pmatrix} 3.7735 & -1 \\ 2.6560 & 7.5463 \end{pmatrix}\begin{pmatrix} \delta x_1 \\ \delta x_2 \end{pmatrix} = -\begin{pmatrix} 0.34886\times10^{-3} \\ 0.16946\times10^{-3} \end{pmatrix}$

Dont la solution est : $\delta\vec{X} = \left[9.03\times10^{-5} \quad 9.25\times10^{-6}\right]^T$

$$\Rightarrow \begin{cases} x_1^5 = x_1^4 + \delta x_1 = 1.3281 \\ x_2^5 = x_2^4 + \delta x_2 = 3.7731 \end{cases}$$

On déduit la convergence de l'algorithme de Newton du fait que les modules de $\delta\vec{X}$ et de $\vec{F}(\vec{X})$ diminuent avec les itérations.

II.3 Méthodes des points fixes en dimension *n*:

Il s'agit d'une solution de : $\vec{X} = \vec{g}(\vec{X})$ (e)

Ou encore de : $\begin{cases} x_1 = g_1(x_1, x_2, x_3, \ldots x_n) \\ x_2 = g_2(x_1, x_2, x_3, \ldots x_n) \\ \vdots \\ x_n = g_n(x_1, x_2, x_3, \ldots x_n) \end{cases}$ (f)

Définition: Tout vecteur $\vec{r}$ solution du système (f) est appelé point fixe de l'application $\vec{g}(\overrightarrow{X})$

L'algorithme de base des méthodes des points fixes en dimension n reste le même:

$$\begin{cases} \overrightarrow{X}^0 \text{ donné} \\ \overrightarrow{X}^{k+1} = \vec{g}(\overrightarrow{X}^k) \end{cases} \qquad \text{où } \overrightarrow{X}^k = \left[x_1^k \; x_2^k \; x_3^k \cdots x_n^k \right]^T$$

Analyse de la convergence:

Définition: le rayon spectrale d'une matrice A est défini par:

$$\rho(A) = \max_{1 \le i \le n} |\lambda_i|$$

Où $|\lambda_i|$ est le module complexe de la valeur propre λ_i de A.

L'équivalent multidimensionnel de la condition $\left|g'(r)\right| < 1$ est tout simplement $\rho(J(\vec{r})) < 1$

Théorème: soit $\vec{r}$, un point fixe de l'application $\overrightarrow{X} = \vec{g}(\overrightarrow{X})$, alors $\vec{r}$ est attractif si $\rho(J(\vec{r})) < 1$; il est répulsif si $\rho(J(\vec{r})) > 1$. Le cas où $\rho(J(\vec{r})) = 1$ est indéterminé.

$\rho(J(\vec{r}))$: rayon spectrale de la matrice jacobienne de $\vec{g}(\overrightarrow{X})$.

$$J(\vec{r}) = \begin{pmatrix} \dfrac{\partial g_1(\vec{r})}{\partial x_1} & \dfrac{\partial g_1(\vec{r})}{\partial x_2} & \cdots & \dfrac{\partial g_1(\vec{r})}{\partial x_n} \\ \dfrac{\partial g_2(\vec{r})}{\partial x_1} & \dfrac{\partial g_2(\overrightarrow{X}^k)}{\partial x_2} & \cdots & \dfrac{\partial g_2(\vec{r})}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \dfrac{\partial g_n(\vec{r})}{\partial x_1} & \dfrac{\partial g_n(\vec{r})}{\partial x_2} & \cdots & \dfrac{\partial g_n(\vec{r})}{\partial x_n} \end{pmatrix}$$

Remarque: Dans le cas général comme dans le cas unidimensionnel, le fait que le point fixe soit attractif ne garantit pas la convergence de l'algorithme (e). On doit toujours s'assurer que la valeur initiale $\overrightarrow{X}^0$ appartient au bassin d'attraction du point fixe.

Exemple: soit $\begin{cases} x_1 = \sqrt{2 - (x_2)^2} \\ x_2 = \sqrt{x_1} \end{cases}$

Cette application ne possède que le seul point fixe $\left[1 \quad 1\right]^T$

Traçage des courbes:

```
ezplot('x1-sqrt(2*x2^2))
hold on
```

```
ezplot('x2-sqrt(x1)')
grid
axis([-6.5 6.5 -5 5])
```

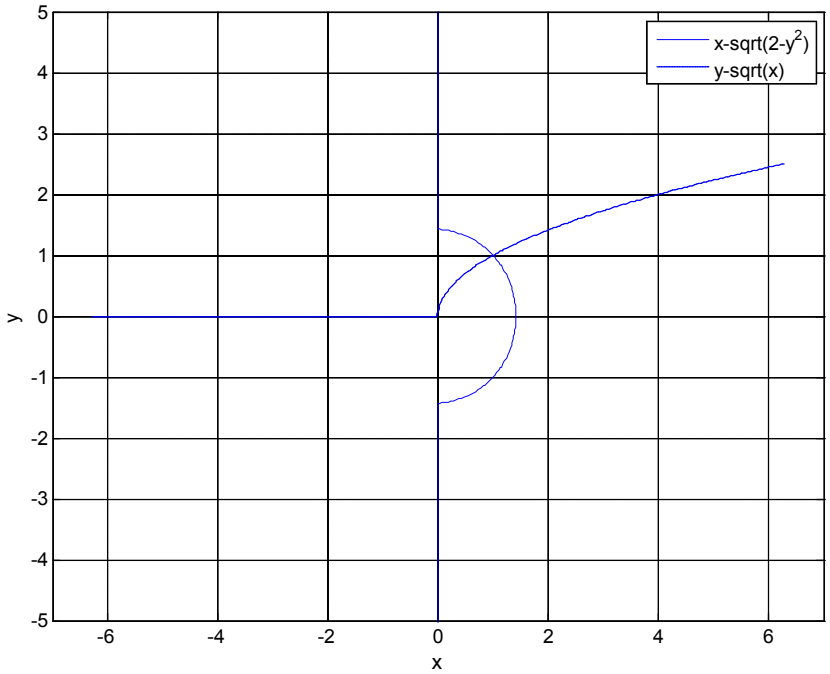

Figure 1.4

$$J(x_1,x_2)=\begin{bmatrix} 0 & \dfrac{-x_2}{\sqrt{2-x_2^2}} \\ \dfrac{1}{2\sqrt{x_1}} & 0 \end{bmatrix}$$

$$J(1,1)=\begin{bmatrix} 0 & -1 \\ \dfrac{1}{2} & 0 \end{bmatrix}$$

Le polynôme caractéristique est alors $p(\lambda)=\lambda^2+\frac{1}{2}$, et les valeurs propres sont $\pm i\sqrt{\frac{1}{2}}$. Le point fixe $\begin{bmatrix}1 & 1\end{bmatrix}^T$ est attractif puisque $\rho(J(1,1))=\sqrt{\frac{1}{2}}<1$

Si $\overline{X}^0=\begin{bmatrix}0 & 0\end{bmatrix}^T$ la solution initiale, on trouve les valeurs suivantes après exécution de l'algorithme (e) :

Méthode des points fixes en dimension 2		
n	x_1^n	x_2^n
1	0.14142×10^{1}	0.00000×10^{0}
2	0.14142×10^{1}	0.11892×10^{1}
3	0.76537×10^{0}	0.11892×10^{1}
4	0.76537×10^{0}	0.87485×10^{0}
5	0.11111×10^{1}	0.87485×10^{0}
	⋮	⋮
23	0.99978×10^{0}	0.10002×10^{1}
24	0.99978×10^{0}	0.99989×10^{0}

On constate une convergence relativement lente vers le point fixe $\begin{bmatrix}1 & 1\end{bmatrix}^T$.

Chapitre 2: Résolution des systèmes d'équations linéaires

2.1 Rappel sur l'algèbre linéaire:

La transposée d'une matrice A ($A \in R^{m \times n}$) est la matrice A^T ($A \in R^{n \times m}$)

$\left(A^T\right)^T = A$; $\left(A+B\right)^T = A^T + B^T$; $\left(A.B.C\right)^T = C^T.B^T.A^T$

$\left(k.A\right)^T = k.A^T$; $\left(A^T\right)^{-1} = \left(A^{-1}\right)^T$

Matrices diagonales et triangulaires:

Matrice diagonale:

Une matrice A telle que $a_{ij} = 0 \;\; si \;\; i \neq j$ s'appelle une matrice diagonale.

$$A = \begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{pmatrix}; \quad \begin{array}{c} Fonction\ Matlab : diag() \\ diag\left(\left[a_{11}\, a_{22} \dots a_{nn}\right]\right) ou\ diag\left(\left[a_{11}\, a_{22} \dots a_{nn}\right], 0\right) \\ \det(A) = a_{11} \times a_{22} \times \dots . a_{nn} \end{array}$$

$$A^{-1} = \begin{pmatrix} \frac{1}{a_{11}} & 0 & \dots & 0 \\ 0 & \frac{1}{a_{22}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{a_{nn}} \end{pmatrix}; \quad A^{-1} \text{ n'existe donc que si } a_{ii} \neq 0 \quad pour\ i = 1, \dots n$$

Matrice triangulaire supérieure:

$$a_{ij} = 0 \;\; pour \;\; i > j, \quad A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{pmatrix}; \quad \begin{array}{c} Fonction\ Matlab : triu() \\ triu(A)\ triu(A,0) \\ \det(A) = a_{11} \times a_{22} \times \dots . a_{nn} \end{array}$$

Son inverse est une matrice triangulaire supérieure.

Le produit de deux matrices triangulaires supérieures est une matrice triangulaire supérieure.

Matrice triangulaire inférieure:

$$a_{ij}=0 \;\; pour \;\; i<j, \quad A=\begin{pmatrix} a_{11} & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n1} & \cdots & a_{nn} \end{pmatrix}; \quad \begin{matrix} Fonction\ Matlab : tril() \\ tril(A)\ ou\ tril(A,0) \\ \det(A)=a_{11}\times a_{22}\times \dots . a_{nn} \end{matrix}$$

Le produit de deux matrices triangulaires inférieures est une matrice triangulaire inférieure.

Matrice identité (unité):

$$\begin{matrix} a_{ii}=1 \\ a_{ij}=0 \;\; \forall\, i\neq j \end{matrix}, \quad I=\begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}; \quad \begin{matrix} Fonction\ Matlab : eye() \\ eye(n)\ ou\ eye([n,n]) \\ \det(A)=a_{11}\times a_{22}\times \dots . a_{nn} \end{matrix}$$

Matrice symétrique: $si\ A^T=A.$ Donc $a_{ij}=a_{ji}, \quad \forall i$ et $\forall j$

Matrice orthogonale: $si\ A^{-1}=A^T$

Matrice définie positive: une matrice A est définie positive (respectivement semi définie positive) si $si\ x^TAx>0, \forall x\neq 0$ (respectivement $x^TAx\geq 0$).

Exemple:

$$Soit\ A=\begin{pmatrix} 1 & 1 \\ 1 & 5 \end{pmatrix}; \quad x=\begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$$

$$Ax=\begin{pmatrix} x_1+x_2 \\ x_1+5x_2 \end{pmatrix}; \quad x^TAx=x_1(x_1+x_2)+x_2(x_1+5x_2)=\left(x_1+x_2\right)^2+4{x_2}^2$$

qui est toujours positif $\forall\, x_1$ et x_2 non nuls.

Les éléments d'une MDP sont positifs.

Les mineurs principaux d'une MDP sont tous positifs.

Matrice de permutation: une telle matrice s'obtient en permutant les lignes et colonnes de la matrice identité. Un seul élément égal à 1 dans chaque ligne et dans chaque colonne. Les autres éléments sont nuls. Quand on multiplie une matrice A à gauche par une matrice de permutation P, on permute les lignes. Si $p_{ij}=1$, la ligne i de PA est la ligne j de A (c.à.d. la ligne i de A est remplacée par la ligne j). Quand la matrice P est placée à droite, on permute les colonnes. Si $p_{ij}=1$, la colonne j de A est remplacée par sa colonne i.

$$\text{Soit } P=\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}; \quad A=\begin{pmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{pmatrix}; PA=\begin{pmatrix} 3 & 4 & 5 \\ 6 & 7 & 8 \\ 0 & 1 & 2 \end{pmatrix}; \quad AP=\begin{pmatrix} 2 & 0 & 1 \\ 5 & 3 & 4 \\ 8 & 6 & 7 \end{pmatrix}$$

Déterminant d'une matrice: Fonction Matlab : det( ). Ex. : det(A)

$$\text{Soit } A=\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix};\quad \det(A)=\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}=a_{11}\times a_{22}-a_{21}\times a_{12}$$

Matrice carrée ($n\times n$) :

$$A=\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & & a_{nn} \end{pmatrix}$$

Si on raye dans A la ligne et la colonne contenant a_{ij} on obtient une matrice à $n-1$ lignes et $n-1$ colonnes notée A_{ij}, son déterminant $\left|A_{ij}\right|=m_{ij}$ s'appelle mineur de a_{ij} dans A.

$$c_{ij}=(-1)^{i+j}\left|A_{ij}\right|=(-1)^{i+j}m_{ij}$$

On appelle c_{ij} cofacteur de a_{ij} : $\det(A)=a_{11}\left|A_{11}\right|-a_{12}\left|A_{12}\right|+a_{13}\left|A_{13}\right|+\ldots+(-1)^{n+1}a_{1n}\left|A_{1n}\right|$

$$=a_{11}m_{11}-a_{12}m_{12}+a_{13}m_{13}+\ldots+(-1)^{n+1}a_{1n}m_{1n}$$

Exemple:

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

$$=a_{11}(a_{22}a_{33}-a_{23}a_{32})-a_{12}(a_{21}a_{33}-a_{23}a_{31})+a_{13}(a_{21}a_{32}-a_{22}a_{31})$$

Propriétés:

$$\det(A)=\det(A)^T$$

$$\det(A.B)=\det(B.A)=\det(A).\det(B)$$

$$\det(A)=\det(A^{-1})=1\quad ;\quad \det(A^{-1})=\frac{1}{\det(A)}$$

Remarques:

Le déterminant concerne les matrices carrées.

Si A est une matrice régulière : $\det(A)\neq 0$. Elle est singulière dans le cas contraire ($\det(A)=0$).

Le déterminant d'une matrice triangulaire ou diagonale est : $\det(A)=\prod_{i=1}^{n}a_{ii}$

Le déterminant se calcule avantageusement pour une décomposition deux matrices triangulaires (LU).

Si A est orthogonale : $\det(A)=\pm 1$

Rang d'une matrice: Fonction Matlab: rank( ). Ex. : rank(A)

C'est la dimension de son plus grand mineur non nul. Si $A^{n\times n}$ est régulière, $rang(A)=n$

Exemple : $A=\begin{pmatrix}1 & 2\\ 2 & 3\end{pmatrix}$; $\det(A)=-1\neq 0\Rightarrow rang(A)=2$

Inverse d'une matrice: Fonction Matlab: inv( ). Ex. : inv(A)

Soit $\det(A)\neq 0$

$$A^{-1}=\frac{\underbrace{\begin{pmatrix}c_{11} & c_{12} & \cdots & c_{1n}\\ c_{21} & a_{22} & \cdots & c_{2n}\\ \vdots & \vdots & \ddots & \vdots\\ c_{n1} & c_{n2} & \cdots & c_{nn}\end{pmatrix}^{T}}_{\text{Matrice adjointe}}}{\det(A)}=\frac{\tilde{A}}{\det(A)}=\frac{adj(A)}{\det(A)}$$

c_{ij} : sont les cofacteurs. $c_{ij}=(-1)^{i+j}m_{ij}$

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

- Une matrice qui possède un inverse s'appelle inversible ou régulière. Une matrice qui n'a pas d'inverse se dénomme singulière. (l'inverse n'est défini que pour les matrices carrées). A^{-1} est unique.
- $(A.B)^{-1}=B^{-1}A^{-1}$

Vecteurs propres et valeurs propres: Fonction Matlab pour les valeurs propres: eig( ). Ex. : eig(A)

Soit $A^{n\times n}$. Le nombre complexe λ est dit valeur propre de A et x vecteur propre de A associé à λ si : $Ax=\lambda x \quad x\neq 0$

Les valeurs propres de A sont solution de l'équation polynomiale : $\det(A-\lambda I)=0$. Ce déterminant est un polynôme de degré n en λ : $P_n(\lambda)=\det(A-\lambda I)$

La matrice A possède donc n valeurs propres qui sont les n racines (réelles ou complexes, simples ou multiples) de ce polynôme.

- **Le polynôme caractéristique** est $P_n(\lambda)$

$$\det(A)=\prod_{i=1}^{n}\lambda_i$$

- **Trace d'une matrice:** $tr(A)=\sum_{i=1}^{n}a_{ii}=\sum_{i=1}^{n}\lambda_i$
- **Norme vectorielle:** Fonction Matlab: norm( )

On appelle P-norme, la norme L_p d'un vecteur $x\in R^n$

$$\left\|\vec{x}\right\|_p = \left(\sum_{i=1}^{n}\left|x_i\right|^p\right)^{\frac{1}{p}} = \left(\left|x_1\right|^p + \left|x_2\right|^p + \ldots \left|x_n\right|^p\right)^{\frac{1}{p}}$$

Norme euclidienne: $\left\|\vec{x}\right\|_2 = \left(\sum_{i=1}^{n}\left|x_i\right|^2\right)^{\frac{1}{2}} = \left(x_1{}^2 + x_2{}^2 + \ldots x_n{}^2\right)^{\frac{1}{2}} = \sqrt{x_1{}^2 + x_2{}^2 + \ldots x_n{}^2}$

Fonction Matlab: norm$\left(\mathrm{x}\right)$ ou norm(x,2)

Norme l_1 : $\left\|\vec{x}\right\|_1 = \sum_{i=1}^{n}\left|x_i\right|$; Fonction Matlab: norm(x,1)

Norme l_∞ : $\left\|\vec{x}\right\|_\infty = \max_{1\le i\le n}\left|x_i\right|$; Fonction Matlab: norm(x,inf)

Norme matricielle: $A \in R^{m\times n}$

$$\left\|A\right\|_p = \left(\sum_{i=1}^{m}\sum_{j=1}^{n}\left|a_{ij}\right|^p\right)^{\frac{1}{p}}$$

$\left\|A\right\|_1 = \max_{1\le j\le n}\sum_{i=1}^{m}\left|a_{ij}\right|$; (sommer en valeur absolue chacune des colonnes de A et choisir la plus grande). Fonction Matlab: norm(A,1)

$\left\|A\right\|_\infty = \max_{1\le j\le m}\sum_{i=1}^{n}\left|a_{ij}\right|$; (sommer en valeur absolue chacune des lignes de A et choisir la plus grande). Fonction Matlab: norm(A,inf)

$\left\|A\right\|_2 = \sqrt{\rho\left(A^T.A\right)}$: $\sqrt{\rho\left(A^T.A\right)}$: rayon spectral de $A^T.A$ (plus grande valeur propre).

Fonction Matlab: ou norm(A) ou norm(A,2).

Rayon spectral d'une matrice A:

Le rayon spectrale d'une matrice A est défini par:

$$\rho(A) = \max_{1\le i\le n}\left|\lambda_i\right|$$

Où $\left|\lambda_i\right|$ est le module complexe de la valeur propre λ_i de A.

- **Fonction vectorielle:**

$$f:\ R^n \to R^n;\quad x \in R^n$$
$$F(x) = \left[f_1(x), f_2(x), \ldots f_n(x)\right]^T$$

- **Gradient de F par rapport à X :**

Soit $\vec{X} = \left[x_1\ x_2\ x_3 \ldots x_n\right]^T$

$$\frac{\partial F(\vec{X})}{\partial \vec{X}} = \frac{\partial F(\vec{X})}{\partial x_i} = \left[\frac{\partial F(\vec{X})}{\partial x_1}, \frac{\partial F(\vec{X})}{\partial x_2}, \ldots \frac{\partial F(\vec{X})}{\partial x_n}\right]^T$$

- **Jacobienne d'une fonction vectorielle:**

$$J(\vec{X}) = \begin{pmatrix} \frac{\partial f_1(\vec{X})}{\partial x_1} & \frac{\partial f_1(\vec{X})}{\partial x_2} & \cdots & \frac{\partial f_1(\vec{X})}{\partial x_n} \\ \frac{\partial f_2(\vec{X})}{\partial x_1} & \frac{\partial f_2(\vec{X})}{\partial x_2} & \cdots & \frac{\partial f_2(\vec{X})}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n(\vec{X})}{\partial x_1} & \frac{\partial f_n(\vec{X})}{\partial x_2} & \cdots & \frac{\partial f_n(\vec{X})}{\partial x_n} \end{pmatrix}$$

2.2 Méthodes directes et indirectes:

Soit le système linéaire $Ax = b$ où A est une matrice carrée $(n \times n)$. x et b sont des vecteurs colonnes. $A \in R^{n \times n}$, $b \in R^n$, $x \in R^n$

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

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{21}x_2 + \ldots + a_{2n}x_n = b_2 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \end{cases} \quad ou \quad \sum_{j=1}^{n} a_{ij}x_j = b_i \;, i = 1:n$$

Résoudre $Ax = b$ c'est de trouver le vecteur x vérifiant le système.

2.2.1 Méthode de Cramer:

Si $\det(A) \neq 0$ (A : matrice régulière) le système admet l'unique solution : $x = x_i$, $i = 1:n$

Où A est la matrice obtenue en remplaçant dans A la i^{ème} colonne par la colonne b.

Numériquement on calcule : $\det(A); \det(A_i), i = 1:n ; \frac{\det(A_i)}{\det(A)}, i = 1:n$

La méthode de Cramer devient très lente dés que n dépasse 4.

Les méthodes de résolution du système $Ax = b$ peuvent être directes ou indirectes (itératives).

Les méthodes directes: elles donnent au bout d'un nombre fini d'opérations une solution exacte du problème. Les méthodes directes sont: méthode de Gauss, méthode de Gauss-Jordan, méthode de décomposition de A en LU, méthode de Cholesky.

Les méthodes itératives (indirectes): une méthode itérative de résolution du système $Ax = b$ consiste à passer au système $x = \alpha x + \beta$ dont la solution est la limite de la suite définie par $x^{(k+1)} = \alpha x^{(k)} + \beta$. $x^{(0)}$ est une approximation initiale.

Les méthodes itératives sont: la méthode de Jacobi, et la méthode de Gauss-Seidel.

2.3 Méthodes directes:

2.3.1 Méthode de Gauss:

Soit $Ax = b$, où A est une matrice $(n \times n)$, régulière. Le principe consiste à transformer une matrice A en matrice triangulaire supérieure.

$$[A:b] \rightarrow [A^*:b^*]$$

$$\text{i.e. : } \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{pmatrix} \rightarrow \begin{pmatrix} a_{11}^* & a_{12}^* & \cdots & a_{1n}^* & b_1^* \\ 0 & a_{22}^* & \cdots & a_{2n}^* & b_2^* \\ 0 & 0 & a_{33}^* & a_{3n}^* & b_3^* \\ 0 & 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & a_{nn}^* & b_n^* \end{pmatrix}$$

Puis on résout le système: $A^* x = b^*$

Algorithme:

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} \qquad i = 1:k,\ j = 1:n$$

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \cdot a_{kj}^{(k)} \qquad i = k+1:n,\ j = 1:n$$

$$\text{et} \quad b_i^{(k+1)} = b_i^{(k)} \qquad i = 1:k$$

$$b_i^{(k+1)} = b_i^{(k)} - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \cdot b_k^{(k)} \qquad i = k+1:n$$

Sous une autre forme : (L_i : représente la ligne i)

$$L_i^{(k+1)} \leftarrow L_i^{(k)} - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \cdot L_k^{(k)} \qquad i = 1:n$$

$$L_i^{(k+1)} \leftarrow L_i^{(k)} - m_{ik}^{(k)} \cdot L_k^{(k)}$$

Résolution par retour arrière :

$$x_n = \frac{1}{a_{nn}^*} b_n^*, \quad x_{n-1} = \frac{1}{a_{n-1n-1}^*} (b_{n-1}^* - a_{n-1n}^* \, x_n)$$

$$x_1 = \frac{1}{a_{11}^*} (b_1^* - a_{12}^* \, x_2 - \ldots . a_{1n}^* \, x_n)$$

$$\det(A) = (-1)^j \prod_{k=1}^{n} a_{kk}^{(k)} \qquad j\text{:est le nombre de permutation}$$

2.3.2 Méthode de Gauss-Jordan:

Soit A une matrice régulière. Le principe consiste à transformer A en une matrice identité.

$$[A:b] \rightarrow [I:b^*]$$

$$Ax = b \Leftrightarrow Ix = b^* \Rightarrow x = b^*$$

Algorithme:

$$a_{kj}^{(k+1)} = \frac{a_{kj}^{(k)}}{a_{kk}^{(k)}} \qquad j = k:n$$

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - a_{ik}^{(k)} \cdot a_{kj}^{(k+1)} \qquad i = 1:n, i \neq k,\ j = k+1:n$$

$$\text{et} \quad b_k^{(k+1)} = \frac{b_k^{(k)}}{a_{kk}^{(k)}} \qquad i = 1:k$$

$$b_i^{(k+1)} = b_i^{(k)} - a_{ik}^{(k)} \cdot b_k^{(k+1)} \qquad i = 1:n, i \neq k$$

Sous une autre forme : (L_i : représente la ligne i)

$$L_k^{(k+1)} \leftarrow \frac{L_k^{(k)}}{a_{kk}^{(k)}}$$

$$L_i^{(k+1)} \leftarrow L_i^{(k)} - a_{ik}^{(k)} \cdot L_k^{(k+1)} \qquad i = 1:n, i \neq k$$

Exemple: résoudre le système suivant par les méthodes de Gauss et Gauss-Jordan.

$$\begin{cases} 2x_1 + 3x_2 - x_3 = 5 \\ 4x_1 + 34 - 3x_3 = 3 \\ -2x_1 + 3x_2 - x_3 = 1 \end{cases}$$

a) Méthode de Gauss:

$$A = \begin{pmatrix} 2 & 3 & -1 \\ 4 & 4 & -3 \\ -2 & 3 & -1 \end{pmatrix};\ b = \begin{pmatrix} 5 \\ 3 \\ 1 \end{pmatrix} \quad ; \quad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

Itération $1: k = 1$; ; $m_{i1}^{(k)}$? $i = 2:3, (i = k+1:n)$

$$m_{21}^{(1)} = \frac{a_{21}^{(1)}}{a_{11}^{(1)}} = \frac{4}{2} = 2\ ;\ m_{31}^{(1)} = \frac{a_{31}^{(1)}}{a_{11}^{(1)}} = \frac{-2}{2} = -1\ ;$$

$$\begin{pmatrix} 2 & 3 & -1 & 5 \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{pmatrix} \begin{matrix} L_2^{(2)} = L_2^{(1)} - m_{21}^{(1)} \cdot L_1^{(1)} \\ L_3^{(2)} = L_3^{(1)} - m_{31}^{(1)} \cdot L_1^{(1)} \end{matrix} \rightarrow \begin{pmatrix} 2 & 3 & -1 & 5 \\ 0 & -2 & -1 & -7 \\ 0 & 6 & -5 & -15 \end{pmatrix}$$

Itération $2: k = 2$; ; $m_{32}^{(2)}$? $i = 3, (i = k+1:n)$

$$m_{32}^{(2)} = \frac{a_{32}^{(2)}}{a_{22}^{(2)}} = \frac{6}{-2} = -3$$

$$\begin{pmatrix} 2 & 3 & -1 & 5 \\ 0 & -2 & -1 & -7 \\ 0 & 6 & -5 & -15 \end{pmatrix} \quad L_3^{(3)} = L_3^{(2)} - m_{32}^{(2)} \cdot L_2^{(2)} \quad \rightarrow \begin{pmatrix} 2 & 3 & -1 & 5 \\ 0 & -2 & -1 & -7 \\ 0 & 0 & -5 & -15 \end{pmatrix}$$

$$A^* x = b^* \Leftrightarrow \begin{pmatrix} 2 & 3 & -1 \\ 0 & -2 & -1 \\ 0 & 0 & -5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 5 \\ -7 \\ -15 \end{pmatrix}$$

$$\Rightarrow \begin{cases} 2x_1+3x_2-x_3=5 \\ -2x_1-x_3=-7 \\ -5x_3=-15 \end{cases} \Leftrightarrow \begin{cases} x_1=1 \\ x_2=2 \\ x_3=3 \end{cases} ; x=\begin{bmatrix}1 & 2 & 3\end{bmatrix}^T$$

b) Méthode de Gauss-Jordan:

Itération $1: k=1$; $a_{kk}^{(k)} = a_{11}^{(1)} = 2 ? \, i = 2:3, (i \neq 1)$

$$\begin{pmatrix} \boxed{2} & 3 & -1 & 5 \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{pmatrix} \quad L_1^{(2)} = \frac{L_1^{(1)}}{a_{11}^{(1)}} = \frac{L_1^{(1)}}{2} \rightarrow \begin{pmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{pmatrix} \begin{matrix} L_2^{(2)} = L_2^{(1)} - a_{21}^{(1)} \cdot L_1^{(2)} \rightarrow \\ L_3^{(2)} = L_3^{(1)} - a_{31}^{(1)} \cdot L_1^{(2)} \end{matrix}$$

$$\begin{pmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & -2 & -1 & -7 \\ 0 & 6 & -2 & 6 \end{pmatrix}$$

Itération $2: k=2$; $a_{kk}^{(k)} = a_{22}^{(2)} = -2 ? \, i = 1:3, (i \neq 2)$

$$\begin{pmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & \boxed{-2} & -1 & -7 \\ 0 & 6 & -2 & 6 \end{pmatrix} \quad L_2^{(3)} = \frac{L_2^{(2)}}{a_{22}^{(2)}} = \frac{L_2^{(2)}}{-2} \rightarrow \begin{pmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 6 & -2 & 6 \end{pmatrix} \begin{matrix} L_1^{(3)} = L_1^{(2)} - a_{12}^{(2)} \cdot L_2^{(3)} \\ \\ L_3^{(3)} = L_3^{(2)} - a_{32}^{(2)} \cdot L_2^{(3)} \end{matrix}$$

$$\begin{pmatrix} 1 & 0 & -5/4 & -11/4 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & -5 & -15 \end{pmatrix}$$

Itération $3: k=3$; $a_{kk}^{(k)} = a_{33}^{(3)} = -5 ? \, i = 1:2, (i \neq 3)$

$$\begin{pmatrix} 1 & 0 & -5/4 & -11/4 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & \boxed{-5} & -15 \end{pmatrix} \quad L_3^{(4)} = \frac{L_3^{(3)}}{a_{33}^{(3)}} = \frac{L_3^{(3)}}{-5} \rightarrow \begin{pmatrix} 1 & 0 & -5/4 & -11/4 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & 1 & 3 \end{pmatrix} \begin{matrix} L_1^{(4)} = L_1^{(3)} - a_{13}^{(3)} \cdot L_3^{(4)} \\ L_2^{(4)} = L_2^{(3)} - a_{23}^{(3)} \cdot L_3^{(4)} \end{matrix}$$

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{pmatrix}}_{I} \Rightarrow x = \begin{bmatrix}1 & 2 & 3\end{bmatrix}^T = b^*$$

2.4 Méthode indirectes (itératives):

2.4.1 Méthode de Jacobi : (méthode itérative)

Soit le système linéaire : $(a_{ii} \neq 0)$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{21}x_2 + \ldots + a_{2n}x_n = b_2 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \end{cases}$$

$$\begin{cases} x_1 = \beta_1 + \alpha_{12}x_2 + \alpha_{13}x_3 + \ldots + \alpha_{1n}x_n \\ x_2 = \beta_2 + \alpha_{21}x_1 + \alpha_{23}x_3 + \ldots + \alpha_{2n}x_n \\ \vdots \\ x_n = \beta_n + \alpha_{n1}x_1 + \alpha_{n2}x_3 + \ldots + \alpha_{nn-1}x_{n-1} \end{cases}$$

Où $\beta_i = \dfrac{b_i}{a_{ii}}, \qquad i = 1:n$ et $\alpha_{ij} = \dfrac{-a_{ij}}{a_{ii}} \quad i \neq j, \quad i, j = 1:n$

$\alpha_{ii} = 0, \qquad i = 1:n$

Ce système réduit sous forme matricielle : $x = \alpha x + \beta$ où $\alpha = (\alpha_{ij})$ et $\beta = (\beta_i)$

$$\alpha = \begin{pmatrix} 0 & \alpha_{12} & \cdots & \alpha_{1n-1} & \alpha_{1n} \\ \alpha_{21} & 0 & \alpha_{23} & \cdots & \alpha_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_{n1} & \alpha_{n2} & \cdots & \alpha_{nn-1} & 0 \end{pmatrix} ; \ \beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{pmatrix}$$

$Ax = b \Leftrightarrow x = \alpha x + \beta$ (forme récursive)

$x^{(0)}$ donné: $x^{(k+1)} = \alpha x^{(k)} + \beta \qquad k \in N$

$$x^{(k+1)} = \alpha x^{(k)} + \beta \Leftrightarrow x_i^{(k+1)} = \sum_{\substack{j=1 \\ j \neq i}}^{n} \alpha_{ij} x_j^{(k)} + \beta_i \ , i = 1:n$$

$$\alpha_{ij} = \frac{-a_{ij}}{a_{ii}} \quad ; \quad \beta_i = \frac{b_i}{a_{ii}}$$

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

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left[b_i - \sum_{j<i}^{n} a_{ij} x_j^{(k)} - \sum_{j>i}^{n} a_{ij} x_j^{(k)} \right] \ ; \ i = 1:n$$

La matrice A peu être écrite sous la forme:

$$A = D + T_i + T_s$$

Où $D = \begin{pmatrix} a_{11} & 0 & 0 & 0 & 0 \\ 0 & a_{22} & 0 & 0 & 0 \\ 0 & 0 & a_{33} & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & a_{nn} \end{pmatrix}$; $T_i = \begin{pmatrix} 0 & 0 & 0 & \ldots & 0 \\ a_{21} & 0 & 0 & \ldots & 0 \\ a_{31} & a_{32} & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ a_{n1} & a_{n2} & \ldots & a_{nn-1} & 0 \end{pmatrix}$; $T_s = \begin{pmatrix} 0 & a_{12} & a_{13} & \ldots & a_{1n} \\ 0 & 0 & a_{23} & \ldots & a_{2n} \\ 0 & 0 & 0 & \ldots & \vdots \\ \vdots & \vdots & \vdots & \ddots & a_{n-1n} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

$$Ax = b \Leftrightarrow (D + T_i + T_s)x = b$$
$$\Rightarrow Dx = -(T_i + T_s)x + b$$
$$x^{(k+1)} = -D^{-1}(T_i + T_s)x^{(k)} + D^{-1}b$$
$$x^{(k+1)} = \alpha_J x^{(k)} + \beta_J$$

Convergence de la méthode de Jacobi:

Une condition nécessaire et suffisante pour que l'algorithme de Jacobi converge indépendamment de $x^{(0)}$ est : $\rho(\alpha_J) < 1$ où $\rho(\alpha) = \max_{1 \le i \le n} |\lambda_i|$ est le rayon spectral. λ_i est la valeur propre de α.

Si la matrice A est à diagonale strictement dominante (DSD) : ($|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| \quad \forall i$) la méthode de Jacobi converge.

Remarque: $\rho(\alpha) = \|\alpha\|$ quelque soit la norme matricielle utilisée.

2.4.2 Méthode de Gauss-Seidel : (méthode indirecte (itérative))

La méthode de Jacobi peut s'écrire aussi comme suit:

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

Dans la méthode de Gauss-Seidel, pour le calcul de $x_i^{(k+1)}$, on peut utiliser $x_1^{(k+1)}, x_2^{(k+1)}, \ldots x_{i-1}^{(k+1)}$, déjà calculés et les $x_{i+1}^{(k+1)}, x_{i+2}^{(k+1)}, \ldots x_n^{(k+1)}$ de l'itération précédente. Cela revient à écrire :

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

Suivant la notation introduite par la méthode de Jacobi, la méthode de Gauss-Seidel s'écrit sous forme matricielle:

$$D x^{(k+1)} = b - T_i x^{(k+1)} - T_s x^{(k)}$$
$$x^{(k+1)} = D^{-1}(b - T_i x^{(k+1)} - T_s x^{(k)})$$
$$(T_i + D)x^{(k+1)} = b - T_s x^{(k)}$$
$$x^{(k+1)} = -(T_i + D)^{-1} T_s x^{(k)} + (T_i + D)b$$
$$x^{(k+1)} = \alpha_G x^{(k)} + \beta_G$$

Cette dernière peut être obtenue comme suit:

1. On garde la première équation inchangée.
2. On remplace dans la 2ième équation $x_1^{(k+1)}$ par sa valeur
3. On remplace dans la 3ième équation $x_1^{(k+1)}$ et $x_2^{(k+1)}$ par leurs valeurs.

Convergence de la méthode de Gauss-Seidel:

Si $\rho(\alpha_G) < 1$ la méthode de Gauss-Seidel converge.

Si la matrice A est à diagonale strictement dominante (DSD), la méthode de Gauss-Seidel converge.

Si A est définie positive, la méthode de Gauss-Seidel converge indépendamment du vecteur initial.

2.4.3 Décomposition LU:

Principe de la méthode:

Le principe consiste à décomposer une matrice A (non singulière) en un produit d'une matrice triangulaire inférieure L (Lower) et une matrice triangulaire supérieure U (Upper).

$$A = LU$$

$$Ax = b$$

$$LUX = b$$

La résolution du système $Ax = b$ se ramène à la résolution de deux systèmes:

D'abord $Ly = b \rightarrow y$?

puis $U\text{x}=\text{y} \rightarrow x$?

Remarque: la décomposition LU n'étant pas unique. Le choix le plus populaire consiste à imposer que la matrice U ait des 1 sur sa diagonale (décomposition de Crout). Matlab, par exemple, préfère de mettre des 1 sur la diagonale de L. Il est en résulte une décomposition LU différente de celle de Crout, mais le principe de base reste le même.

2.4.3.1 Décomposition de Crout:

On considère une matrice 4×4

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} = \begin{pmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{pmatrix} \times \begin{pmatrix} 1 & u_{12} & u_{13} & u_{14} \\ 0 & 1 & u_{23} & u_{24} \\ 0 & 0 & 1 & u_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Identification:

1) Produit des lignes de L par la première colonne de U :

$$l_{11} = a_{11}, \quad l_{21} = a_{21}, \quad l_{31} = a_{31}, \quad l_{41} = a_{41}$$

(1^{ere} colonne de L est égale à la 1^{ere} colonne de A).

2) Produit de la $1^{ère}$ ligne de L par les colonnes de U :

$$l_{11}u_{12} = a_{12}, \quad l_{11}u_{13} = a_{13}, \quad l_{11}u_{14} = a_{14}$$
$$\Rightarrow u_{12} = \frac{a_{12}}{l_{11}}, \quad u_{13} = \frac{a_{13}}{l_{11}}, \quad u_{14} = \frac{a_{14}}{l_{11}}$$

On a la première ligne de U si $l_{11} \neq 0$

3) Produit des lignes de L par la 2ième colonne de U :

$$\begin{cases} l_{21}u_{12}+l_{22}=a_{22} \\ l_{31}u_{12}+l_{32}=a_{32} \\ l_{41}u_{12}+l_{42}=a_{42} \end{cases} \Rightarrow \begin{cases} l_{22} = a_{22} - l_{21}u_{12} \\ l_{32} = a_{32} - l_{31}u_{12} \\ l_{42} = a_{42} - l_{41}u_{12} \end{cases}$$

4) Produit de la 2ième ligne de L par les colonnes de U :

$$\begin{cases} l_{21}u_{13} + l_{22}u_{23} = a_{23} \\ l_{21}u_{14} + l_{22}u_{24} = a_{24} \end{cases} \Rightarrow \begin{cases} u_{23} = \dfrac{a_{23} - l_{21}u_{13}}{l_{22}} \\ u_{24} = \dfrac{a_{24} - l_{21}u_{14}}{l_{22}} \end{cases}$$

5) Produit des lignes de L par la 3ième colonne de U :

$$\begin{cases} l_{31}u_{13} + l_{32}u_{23} + l_{33} = a_{33} \\ l_{41}u_{13} + l_{42}u_{23} + l_{43} = a_{43} \end{cases} \Rightarrow \begin{cases} l_{33} = a_{33} - l_{31}u_{13} - l_{32}u_{23} \\ l_{43} = a_{43} - l_{41}u_{13} - l_{42}u_{23} \end{cases}$$

6) Produit de la ligne 3 de L et la 4ième colonne de U :

$$l_{31}u_{14} + l_{32}u_{24} + l_{33}u_{34} = a_{34}$$
$$\Rightarrow u_{34} = \frac{a_{34} - l_{31}u_{14} - l_{32}u_{24}}{l_{33}}$$

7) Produit de la ligne 4 de L et la 4ième colonne de U :

$$l_{41}u_{14} + l_{42}u_{24} + l_{43}u_{34} + l_{44} = a_{44}$$
$$\Rightarrow l_{44} = a_{44} - l_{41}u_{14} - l_{42}u_{24} - l_{43}u_{34}$$

Algorithme : Décomposition de Crout:

Décomposition LU (sans permutation de lignes)

1ère colonne de L : $l_{i1} = a_{i1} \qquad i = 1, 2, ...n$

1ère ligne de U : $u_{1i} = \dfrac{a_{1i}}{l_{11}} \qquad i = 1, 2, ...n$

pour $i = 2, ...n-1$

Calcul du pivot : $l_{ii} = a_{ii} - \sum_{k=1}^{i-1} l_{ik}u_{ki}$

pour $j = i+1, i+2...n$

Calcul de la i^{ème} colonne de L

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

Calcul de la i^{ème} colonne de U

$$U_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj}}{l_{ii}}$$

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

2. Descente et remontée triangulaires

Descente triangulaire pour résoudre

$$Ly = b \; ; \quad y_1 = \frac{b_1}{l_{11}}$$

$$pour \;\; i = 2,...n$$

$$y_i = \frac{b_i - \sum_{k=1}^{i-1} l_{ik} y_k}{l_{ii}}$$

Remonté triangulaire pour résoudre

$$Ux = y \quad (u_{ii} = 1)$$

$$x_n = y_n$$

$$pour \;\; i = n-1, n-2, ...2,1$$

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

Dans cet algorithme $l_{ii} \neq 0$

Remarque: *si* $A = LU$ avec

L=(l_{ij}) *avec* $l_{ii} = 1 \quad (1 \leq i \leq n)$ une matrice triangulaire inférieure

et U une matrice triangulaire sup*érieure*

L et U sont données par:

$$U_{1j} = a_{1j} \quad pour \;\; j = 1,..n$$

$$l_{i1} = \frac{a_{i1}}{u_{11}} \quad pour \; i = 1,..n; \; l_{ii} = 1$$

$$l_{ij} = 0 \quad pour \;\; j = i+1,..n$$

$$u_{ij} = 0 \quad pour \;\; j = 1,..i-1$$

$$u_{ij}=a_{ij}-\sum_{k=1}^{i-1} l_{ik}u_{kj}$$

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

Exemple:

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

$$L=\begin{pmatrix}3 & 0 & 0\\ 1 & l_{22} & 0\\ 2 & l_{32} & l_{33}\end{pmatrix}$$

1ère ligne de U ? Pivot=3. On divise la ligne 1 de A par 3:

$$U=\begin{pmatrix}1 & -1/3 & 2/3\\ 0 & 1 & u_{23}\\ 0 & 0 & 1\end{pmatrix}\quad ; u_{12}=\frac{a_{12}}{l_{11}}=\frac{-1}{3}\quad ;\quad u_{13}=\frac{a_{13}}{l_{11}}=\frac{2}{3}$$

2ième colonne de L:

$$l_{ii}=a_{ii}-\sum_{k=1}^{i-1} l_{ik}u_{ki}\,;\, l_{22}=a_{22}-\sum_{k=1}^{1} l_{21}u_{12}=a_{22}-l_{21}u_{12}=2-(1)(-\frac{1}{3})=\frac{7}{3}$$

$$l_{32}=a_{32}-l_{31}u_{12}=2-(2)(-\frac{1}{3})=-\frac{4}{3}\Rightarrow L=\begin{pmatrix}3 & 0 & 0\\ 1 & 7/3 & 0\\ 2 & -4/3 & l_{33}\end{pmatrix}$$

2ième ligne de U:

$$U_{ij}=\frac{a_{ij}-\sum_{k=1}^{i-1} l_{ik}u_{kj}}{l_{ii}}\,;U_{23}=\frac{a_{23}-l_{21}u_{13}}{l_{22}}=\frac{3-(-1)\left(\frac{2}{3}\right)}{\frac{7}{3}}=1$$

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

l_{33}?

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

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

$Ly = b$: descente triangulaire

$$y_1 = \frac{b_1}{l_{11}} = \frac{12}{3} = 4\,; \quad y_2 = \frac{b_2 - l_{21}y_1}{l_{22}} = \frac{11-(1)(4)}{\frac{7}{3}} = 3\,;$$

$$y_3 = \frac{b_3 - l_{31}y_1 - l_{32}y_2}{l_{33}} = \frac{2-(2)(4)-\left(\frac{-4}{3}\right)(3)}{(-1)} = 2$$

Résolution de $ux = y$

$$x_3 = y_3 = 2\,; \quad x_2 = y_2 - u_{23}x_3 = 3-(1)(2) = 1\,; \quad x_1 = y_1 - u_{12}x_2 - u_{13}x_3 = 4 - \left(\frac{-1}{3}\right)(1) - \left(\frac{2}{3}\right)2 = 3$$

$$x = \begin{bmatrix} 3 & 1 & 2 \end{bmatrix}^T$$

Remarque: il est possible d'utiliser la méthode d'élimination de Gauss pour résoudre le système $Ax = b$; $A = LU$

$$L = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ m_{21} & 1 & 0 & \dots & 0 \\ m_{31} & m_{32} & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & m_{n3} & & 1 \end{pmatrix}; B = \begin{pmatrix} a_{1n+1}^{(1)} \\ a_{2n+1}^{(2)} \\ a_{3n+1}^{(3)} \\ \vdots \\ a_{nn+1}^{(n)} \end{pmatrix}; U = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & a_{nn}^{(n)} \end{pmatrix}$$

Algorithme:

pour $i = k+1 : n$

$$m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$$

$$a_{ik} = m_{ik}$$

pour $j = k+1 : n+1$

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik}a_{kj}^{(k)}$$

Fin pour

Fin pour

Notation en forme compacte:

$$\begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & & \cdots & a_{1n}^{(1)} & a_{1n+1}^{(1)} \\ m_{21} & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} & a_{2n+1}^{(2)} \\ m_{31} & m_{32} & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} & a_{3n+1}^{(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ m_{n1} & m_{n2} & m_{n3} & \cdots & a_{nn}^{(n)} & a_{nn+1}^{(n)} \end{pmatrix}$$

Exemple: utiliser l'élimination de Gauss pour construire une factorisation LU pour la matrice suivante: $A=\begin{pmatrix} 4 & 3 & -1 \\ -2 & -4 & 5 \\ 1 & 2 & 6 \end{pmatrix}$

$$k=1\,;\, m_{21}=\frac{-2}{4}=\frac{-1}{2}\,;\, m_{31}=\frac{1}{4}$$

$$A=\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\times\begin{pmatrix} 4 & 3 & -1 \\ -2 & -4 & 5 \\ 1 & 2 & 6 \end{pmatrix}\begin{matrix} \\ L_2-m_{21}L_1 \\ L_3-m_{31}L_1 \end{matrix}$$

$$\Rightarrow A=\begin{pmatrix} 1 & 0 & 0 \\ -0.5 & 1 & 0 \\ 0.25 & 0 & 1 \end{pmatrix}\times\begin{pmatrix} 4 & 3 & -1 \\ 0 & -2.5 & 4.5 \\ 0 & 1.25 & 6.25 \end{pmatrix}$$

$$k=2\,;\, m_{32}=\frac{1.25}{-2.5}=-0.5$$

$$A=\begin{pmatrix} 1 & 0 & 0 \\ -0.5 & 1 & 0 \\ 0.25 & 0 & 1 \end{pmatrix}\times\begin{pmatrix} 4 & 3 & -1 \\ 0 & -2.5 & 4.5 \\ 0 & 1.25 & 6.25 \end{pmatrix}\begin{matrix} \\ \\ L_3-m_{32}L_2 \end{matrix}$$

$$\Rightarrow A=\underbrace{\begin{pmatrix} 1 & 0 & 0 \\ -0.5 & 1 & 0 \\ 0.25 & -0.5 & 1 \end{pmatrix}}_{L}\times\underbrace{\begin{pmatrix} 4 & 3 & -1 \\ 0 & -2.5 & 4.5 \\ 0 & 0 & 8.5 \end{pmatrix}}_{U}$$

Décomposition LU en Matlab:

$[L,U]=lu(A);$

U : matrice triangulaire supérieure.

L : matrice triangulaire inférieure permutée telle que : $A=L*U$

$[L,U,P]=lu(A);$

U : matrice triangulaire supérieure.

L : matrice triangulaire inférieure ayant des 1 sur la diagonale telle que : $L*U=P*A$

Exemple:

$A=[1\quad 2\quad 3;4\quad 5\quad 6;7\quad 8\quad 0];$

$[L1,U]=lu(A)$

$$L1=\begin{pmatrix} 0.1429 & 1.0000 & 0 \\ 0.5741 & 0.5 & 1.0000 \\ 1.0000 & 0 & 0 \end{pmatrix}\quad ;\quad U=\begin{pmatrix} 7 & 8 & 0 \\ 0 & 0.8571 & 3 \\ 0 & 0 & 4.5 \end{pmatrix}$$

Notons que $L1$ est une permutation d'une matrice triangulaire inférieure.

Si on permute les lignes 2 et 3, puis 1 et 2, on obtient une matrice triangulaire inférieure ayant des 1 dans la diagonale.

$L1*U$ donne A ; $x = inv(U)*inv(L1)*b$

$[L2, U, P] = lu(A)$

$$L2 = \begin{pmatrix} 1 & 0 & 0 \\ 0.1429 & 1 & 0 \\ 0.5741 & 0.5 & 1 \end{pmatrix}; \quad U = \begin{pmatrix} 7 & 8 & 0 \\ 0 & 0.8571 & 3 \\ 0 & 0 & 4.5 \end{pmatrix}; \quad P = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Notons que $L2 = P*L1$

$$P*A - L2*U = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Dans ce cas on a:

$$P*A = L2*U \Rightarrow A = P^{-1}L2U = P^{T}L2U$$
$$Ax = b \Leftrightarrow P^{T}L2Ux = b$$
$$P^{-1}L2Ux = b$$
$$L2Ux = Pb$$
$$L2y = Pb \Rightarrow y ?$$
$$Ux = y \Rightarrow x ?$$

2.4.3.2 Méthode de Cholesky:

Soit A une matrice carrée d'ordre n non singulière et symétrique ($A = A^{T}$). Cette méthode consiste à décomposer A en deux matrices L et L^{T} de même dimension que A (L : matrice triangulaire inférieure). Pour que L existe, il faut que A soit définie positive.

Remarque: L n'est pas unique.

Cette décomposition devient unique si l'on fixe à l'avance les éléments diagonaux l_{ii} avec $l_{ii} > 0$

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

Calcul des l_{ii} : $A = LL^{T}$ (Par identification)

$$\begin{cases} l_{11}^2 = a_{11} \\ l_{i1}^2 + l_{i2}^2 + l_{i3}^2 + ... + l_{ii-1}^2 + l_{ii}^2 = a_{ii} \\ l_{i1}l_{ji} + l_{i2}l_{j2} + l_{i3}l_{j3} + ... + l_{ij}l_{jj} = a_{ij}, \quad i > j \\ l_{ij} = 0, \quad si\ i < j \end{cases}$$

$$\begin{cases} l_{11} = \sqrt{a_{11}} \\ l_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2}\,, \quad i = 2:n \\ l_{ij} = \frac{1}{l_{jj}}(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk})\,, \quad i > j \ (i = j+1:n) \\ l_{ij} = 0\,, \quad si\ i < j \end{cases}$$

$$Ax = b$$

Résolution de $LL^T x = b \quad \begin{cases} Ly = b \\ L^T x = y \end{cases}$

D'où l'on déduit : $\begin{cases} y_1 = \frac{b_1}{l_{11}} \\ y_i = \frac{b_i - \sum_{k=1}^{i-1} l_{ik} y_k}{l_{ii}} \quad i = 2:n \end{cases}$

$$\begin{cases} x_n = \frac{y_n}{l_{nn}} \\ x_i = \frac{y_i - \sum_{k=i+1}^{n} l_{ki} x_k}{l_{ii}} \quad i = 1:n-1 \end{cases}$$

Remarque: la méthode de Cholesky permet le calcul du déterminant de A. En effet:

$$A = LL^T \Rightarrow \det(A) = \det(LL^T) = \det(L)\det(L) = (\det(L))^2 = (\prod_{i=1}^{n} l_{ii})^2$$

Décomposition de Cholesky en Matlab:

$l = chol(A, 'lower')$

A est une matrice définie positive et symétrique.

Exemple:

$A = [2 \quad 3 \quad 4; 3 \quad 5 \quad 6; 4 \quad 6 \quad 9];$

$l = chol(A, 'lower')$

$$l = \begin{pmatrix} 1.4142 & 0 & 0 \\ 2.1213 & 0.7071 & 0 \\ 2.8284 & 0 & 1 \end{pmatrix}$$

2.4.4 Décomposition QR:

Il existe différents moyens de décomposer une matrice quelconque. L'une des plus utilisée est la décomposition QR.

Soit A une matrice de $R^{m \times n}$. A peut être factorisée comme suit : $A = QR$

Q: matrice orthogonale ($m \times n$), $Q^T Q = I$.

R: matrice triangulaire supérieure.

On peut utiliser cette décomposition pour résoudre un système linéaire (problème des moindres carrées) : $Ax = b \quad (A^T Ax = A^T b)$.

$$\text{Si } A = QR \text{ et } Q^T Q = I$$

$$A^T Ax = A^T b \Leftrightarrow (QR)^T QRx = (QR)^T b$$
$$R^T Q^T QRx = R^T Q^T b \Leftrightarrow R^T Rx = R^T Q^T b$$
$$\Leftrightarrow Rx = Q^T b \quad (R: \text{non singulière})$$

A: n'est pas forcément carrée (système surdéterminé).

$$QQ^T = I \quad \text{(pour les matrices carrées).}$$

$$Q^{-1} = Q^T; \det(Q) = \pm 1; \|Qx\| = \|x\|; (Qx)^T (Qy) = x^T y$$

La multiplication des vecteurs avec une matrice orthogonale préserve les normes et les produits scalaires.

Algorithme: (résoudre un problème des moindres carrées par la décomposition QR)

1. Calculer la décomposition QR . $A = QR$
2. Calculer $d = Q^T b$
3. Résoudre $Rx = d$ par retour arrière.

Exemple:

$$A = \begin{pmatrix} 3 & -6 \\ 4 & -8 \\ 0 & 1 \end{pmatrix}; b = \begin{pmatrix} -1 \\ 7 \\ 2 \end{pmatrix}$$

1. $A = QR$ avec:

$$Q = \begin{pmatrix} 3/5 & 0 \\ 4/5 & 0 \\ 0 & 1 \end{pmatrix} \quad ; \quad R = \begin{pmatrix} 5 & -10 \\ 0 & 1 \end{pmatrix}$$

2. $d = Q^T b = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$

3. Résoudre: $\begin{pmatrix} 5 & -10 \\ 0 & 1 \end{pmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$

D'où $x_1 = 5; x_2 = 2$

Il existe plusieurs méthodes pour le calcul de la décomposition QR : méthode de Gram-Schmidt, méthode de Housholder,…etc.

2.4.4.1 Méthode de Gram-Schmidt:

$Soit \quad A = \left[a_1 \middle| a_2 \ldots \middle| a_n \right]$

$a_1, a_2, \ldots a_n$: vecteurs colonnes

$$u_1 = a_1 \; ; \; e_1 = \frac{u_1}{\|u_1\|}$$

$$u_2 = a_2 - (a_2^T \times e_1) e_1 \; ; \; e_2 = \frac{u_2}{\|u_2\|}$$

$$u_{k+1} = a_{k+1} - (a_{k+1}^T \times e_1) e_1 - \ldots (a_{k+1}^T \times e_k) e_k \; ; \; e_{k+1} = \frac{u_{k+1}}{\|u_{k+1}\|}$$

$\| \; \|$: *la norme* L_2

Décomposition QR **:**

$$A = \left[a_1 \middle| a_2 \ldots \middle| a_n \right] = \left[e_1 \middle| e_2 \ldots \middle| e_n \right] \times \begin{pmatrix} a_1^T \times e_1 & a_2^T \times e_1 & \ldots & a_n^T \times e_1 \\ 0 & a_2^T \times e_1 & \ldots & a_n^T \times e_1 \\ \vdots & 0 & & \vdots \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \ldots & a_n^T \times e_n \end{pmatrix}$$

Exemple:

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} ; \quad a_1 = [1,1,0]^T \; ; \; a_2 = [1,0,1]^T \; ; \; a_1 = [0,1,1]^T$$

$$u_1 = a_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} ; e_1 = \frac{u_1}{\|u_1\|} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}$$

$$u_2 = a_2 - (a_2^T \times e_1) e_1 = [1,0,1]^T - \frac{1}{\sqrt{2}} \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0 \right]^T = \left[\frac{1}{2}, \frac{-1}{2}, 1 \right]^T ;$$

$$e_2 = \frac{u_2}{\|u_2\|} = \frac{1}{\sqrt{\frac{3}{2}}} \left[\frac{1}{2}, \frac{-1}{2}, 1 \right]^T = \left[\frac{1}{\sqrt{6}}, \frac{-1}{\sqrt{6}}, \frac{2}{\sqrt{6}} \right]^T$$

$$u_3 = a_3 - (a_3^T \times e_1) e_1 - (a_3^T \times e_2) e_2 = \left[\frac{-1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}} \right]^T ; e_3 = \frac{u_3}{\|u_3\|} = \left[\frac{-1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}} \right]^T$$

$$\text{Ainsi } Q=\begin{pmatrix} a_1^T\times e_1 & a_2^T\times e_1 & a_3^T\times e_1 \\ 0 & a_2^T\times e_2 & a_3^T\times e_2 \\ 0 & 0 & a_3^T\times e_3 \end{pmatrix}=\begin{pmatrix} \frac{2}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & \frac{3}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ 0 & 0 & \frac{2}{\sqrt{3}} \end{pmatrix}$$

Factorisation QR en MATLAB: $qr()$

$$\text{Soit } A=\begin{pmatrix} 2 & 3 \\ 1 & -1 \\ 2 & 2 \\ 4 & 1 \end{pmatrix}; b=\begin{pmatrix} 7 \\ 1 \\ 5 \\ 6 \end{pmatrix}$$

$A=[2 \quad 3;1 \quad -1;2 \quad 2;4 \quad 1]; b=[7;1;5;6];$

$[Q,R]=qr(A)$

$disp('Q='); disp(Q)$

$disp('R='), disp(R),$

$d=Q^T b;$

$x=R(1:2,1:2)\backslash d(1:2)$

$x=inv((R(1:2,1:2))*d(1:2)$

$$Q=\begin{pmatrix} -0.4 & 0.6828 & -0.4756 & -0.3842 \\ -0.2 & -05295 & 0.0714 & -0.8213 \\ -0.4 & 0.3344 & 0.8522 & -0.0441 \\ -0.8 & -0.3762 & -0.2061 & 0.4195 \end{pmatrix}; \quad R=\begin{pmatrix} -5 & -2.6 \\ 0 & 2.8705 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}; \quad x=[1.2961, 1.2767]^T$$

2.4.5 Décomposition en valeurs singulières SVD (Singular value decomposition):

La décomposition en valeurs singulières (SVD : Singular value decomposition) exprime la matrice $A^{m\times n}$ sous la forme :

$$A=U.S.V^T$$

$U^{m\times m}$ et $V^{n\times n}$ Sont des matrices orthogonales.

$S^{m\times n}$ est une matrice diagonale (de même taille que A). Ses termes diagonaux, tous positifs ou nuls, sont les valeurs singulières de A . Le rang de A est égal au nombre de valeurs singulières non nulles.

- si A est régulière tous les s_{ii} sont >0
- La matrice inverse est donnée par:

$$A^{-1}=(U.S.V^T)^{-1}=(V^T)^{-1}(S^{-1}).(U^{-1})=V.(S^{-1}).U^T$$

(L'inverse d'une matrice orthogonale est égal à sa transposée).

D'où on déduit la solution du système :

$$x = A^{-1}b = V.(S^{-1}).U^T * b = V.diag(\frac{1}{S_{ii}}).U^T * b$$

Décomposition SVD en Matlab:

Soit:

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}; A = [1 \quad 2;3 \quad 4;5 \quad 6]$$

$$[U, S, V] = svd(A)$$

$$U = \begin{pmatrix} -0.2298 & 0.8835 & 0.4082 \\ -0.5247 & 0.2408 & -0.8165 \\ -0.8196 & -0.4019 & 0.4082 \end{pmatrix}$$

$$S = \begin{pmatrix} 9.5255 & 0 \\ 0 & 0.5143 \\ 0 & 0 \end{pmatrix} \quad ; \quad V = \begin{pmatrix} -0.6196 & -0.7849 \\ -0.7849 & 0.6196 \end{pmatrix}$$

$$err = (U * S * V') - A$$

$$= 1.0e-14 * [-0.0666 \quad -0.0888; -0.0888 \quad -0.1332; -0.3553 \quad -0.2665]$$

Chapitre 3: Différentiation et intégration numériques

3.1 Introduction:

Le problème consiste à obtenir des approximations des différentes dérivées d'une fonction $f(x)$ de même que de $\int_{x_0}^{x_n} f(x)dx$. On parle alors de dérivation numérique et d'intégration numérique. On fait face à ce type de problèmes lorsque, par exemple, on connait la position d'une particule à intervalles de temps réguliers et que l'on souhaite obtenir sa vitesse. On doit alors effectuer la dérivée de la position connue seulement en quelques points. De même, l'accélération de cette particule nécessite le calcul de la dérivée seconde. Si, à l'inverse, on connait la vitesse d'une particule à certains intervalles de temps, on obtient la distance inconnue en intégrant la vitesse dans l'intervalle $[x_0, x_n]$.

3.2 Différentiation numérique:

3.2.1 Dérivées d'ordre 1:

Le processus numérique pour approximer la dérivée de $f(x)$ est le suivant:

$$f'(x) = \lim_{h\to 0} \frac{f(x+h)-f(x)}{h} \qquad (3.1)$$

Cette méthode consiste à choisir une séquence $\{h_k\}$ tel que $h_k \to 0$ et calculer la limite de la séquence:

$$D_k = \lim_{h\to 0} \frac{f(x+h_k)-f(x)}{h_k} \qquad k=1,2,... \qquad (3.2)$$

On calcule un nombre fini des termes $D_1, D_2, ...D_N$ dans la séquence (3.2) jusqu'à ce que $|D_{N+1}-D_N| \geq |D_N-D_{N-1}|$

Exemple:

Soit $f(x)=e^x$ et $x=1$. Calculer les quotients de différence D_k en utilisant les pas $h_k = 10^{-k}$ pour $k = 1,2,...10$. Utiliser 9 décimales pour les calculs.

$$D_k = \frac{f(x+h)-f(x)}{h_k} = \frac{(e^{1+h_k}-e)}{h_k}$$

h_k	$f_k = f(1+h_k)$	$f_k - e$	$D_k = (f_k - e)/h_k$
$h_1 = 0.1$	3.004166024	0.285884196	2.858841960
$h_2 = 0.1$			2.731918700
$h_3 = 0.1$			2.719642000
$h_4 = 0.1$			2.718420000
$h_5 = 0.1$			2.718300000
$h_6 = 10^{-6}$			2.719000000
$h_7 = 10^{-7}$			2.720000000
$h_8 = 10^{-8}$			2.8000000
$\vdots$			3.000000
$h_{10} = 10^{-10}$			0.000000

La valeur exacte est $f'(1) \approx 2.718281828$.

La valeur $h_k = 10^{-5}$ donne la meilleure approximation : $D_5 = 2.7183$

On a $0.0007 = |D_6 - D_5| \geq |D_5 - D_4| = 0.00012$

$\Rightarrow D_5$ est la meilleure approximation.

Pour $h_1 = 0.1$ (trop large), on n'obtient pas une bonne approximation.

Si h_k est très petit $f(x+h)$ et $f(x)$ f(x) sont très proches : on soustrait de quantités presque égales $\Rightarrow$ la différence tend vers 0(le cas de $h_{10}, D_{10} \to 0$).

Formules de différences finies d'ordre 1 pour $f'(x)$:

$$f'(x) = \frac{f(x+h)-f(x)}{h} + O(h)$$: Différence avant d'ordre 1

$$f'(x) = \frac{f(x)-f(x-h)}{h} + O(h)$$: Différence arrière d'ordre 1

Formules de différences finies d'ordre 2 pour $f'(x)$:

$$f'(x) = \frac{-f(x+2h)+4f(x+h)-3f(x)}{2h} + O(h^2)$$: Différence avant d'ordre 2

$$f'(x) = \frac{f(x+h) - f(x-h) - 3f(x)}{2h} + O(h^2)$$: Différence centrée d'ordre 2

$$f'(x) = \frac{3f(x) - 4f(x-h) + f(x-2h)}{2h} + O(h^2)$$: Différence arrière d'ordre 2

3.2.2 Dérivées d'ordre supérieur:

Formules de différences finies pour $f''(x)$:

$$f''(x) = \frac{f(x-2h) - 2f(x-h) + f(x)}{h^2} + O(h)$$: Différence arrière d'ordre 1

$$f''(x) = \frac{f(x+2h) - 2f(x+h) + f(x)}{h^2} + O(h)$$: Différence avant d'ordre 1

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + O(h^2)$$: Différence centrée d'ordre 2

$$f''(x) = \frac{-f(x+2h) + 16f(x+h) - 30f(x) + 16f(x-h) - f(x-2h)}{12h^2} + O(h^4)$$

Différence centrée d'ordre 4

Formules de différences finies pour $f^{(4)}(x)$:

$$f^{(4)}(x) = \frac{f(x+2h) - 4f(x+h) + 6f(x) - 4f(x-h) + f(x-2h)}{h^4} + O(h^2)$$

Différence centrée d'ordre 2

3.3 Intégration numérique:

3.3.1 Méthode du trapèze (simple): (MTS)

On souhaite évaluer $\int_{x_0}^{x_n} f(x)dx$ où $f(x)$ est une fonction connue seulement en deux points ou encore une fonction n'ayant pas de primitives. Une solution consiste à remplacer $f(x)$ par le polynôme de degré 1 passant par les points $(x_0, f(x_0))$ et $(x_1, f(x_1))$ comme l'illustre la Figure 3.1. La valeur approximative de l'intégrale correspond à l'aire sous la courbe du polynôme. Cette aire forme un trapèze.

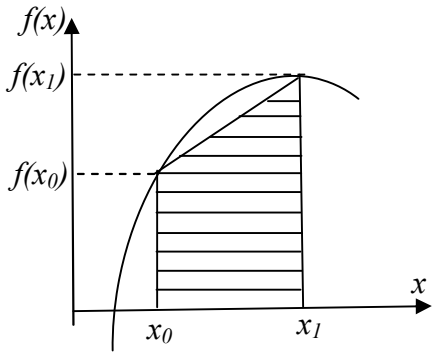

Figure 3.1

L'intégration numérique est un outil essentiel utilisé par les ingénieurs et les scientifiques pour donner des réponses approximatives aux intégrales difficile à résoudre analytiquement.

$$\int_{x_0}^{x_1} f(x)dx \approx \frac{x_1 - x_0}{2}(f(x_0) + f(x_1)) = \frac{h}{2}(f(x_0) + f(x_1))$$

Exemple : évaluer numériquement $\int_0^{\frac{\pi}{2}} \sin(x)dx$ par la méthode du trapèze.

$$I = \int_0^{\frac{\pi}{2}} \sin(x)dx \approx \frac{\frac{\pi}{2}}{2}(\sin(0) + \sin(\frac{\pi}{2})) = \frac{\pi}{4} = 0.785398164$$

Qui est une piètre approximation de la valeur exacte de 1. L'approximation est assez médiocre.

3.3.2 Méthode des trapèzes composée:

On décompose l'intervalle $[a,b]$ en n sous-intervalles de longueur $h = \frac{b-a}{n}$.

Dans chaque sous-intervalle $[x_i, x_{i+1}]$, on utilise la méthode du trapèze.

$$\int_a^b f(x)dx = \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} f(x)dx \approx \sum_{i=1}^{n} \frac{h}{2}(f(x_{i-1}) + f(x_i)) = \sum_{i=0}^{n-1} \frac{h}{2}(f(x_i) + f(x_{i+1}))$$

$$= \frac{h}{2}\left[(f(x_0) + f(x_1)) + (f(x_1) + f(x_2)) + \ldots + (f(x_{n-2}) + f(x_{n-1})) + (f(x_{n-1}) + f(x_n))\right]$$

Tous les termes $f(x_i)$ sont répétés 2 fois sauf le premier et le dernier.

$$\int_a^b f(x)dx = \frac{h}{2}\left(f(x_0) + 2\left[f(x_1) + f(x_2)) + \ldots + (f(x_{n-2}) + f(x_{n-1})\right] + f(x_n)\right)$$

$$x_i = x_0 + ih$$

$$x_i = a + ih, \quad i = 0,1\ldots n$$

Exemple: évaluer $\int_0^{\frac{\pi}{2}} \sin(x)dx$ par la méthode des trapèzes composée en utilisant:

a) 4 intervalles. b) 8 intervalles.

$$a)\ h = \frac{\frac{\pi}{2} - 0}{4} = \frac{\pi}{8}$$

$$I = \int_0^{\frac{\pi}{2}} \sin(x)dx \approx \frac{\frac{\pi}{8}}{2}(\sin(0) + 2\times(\sin(\frac{\pi}{8}) + \sin(\frac{\pi}{4}) + \sin(\frac{3\pi}{8})) + \sin(\frac{\pi}{2})) = 0.9871158$$

L'erreur absolue par rapport à la solution exacte 1 est 0.01288. On constate une nette amélioration en comparaison avec le résultat obtenu avec un seul intervalle.

$$b)\ h = \frac{\frac{\pi}{2} - 0}{8} = \frac{\pi}{16}$$

$$I = \int_0^{\frac{\pi}{2}} \sin(x)dx \approx \frac{\frac{\pi}{16}}{2}(\sin(0) + 2\times(\sin(\frac{\pi}{16}) + \sin(\frac{\pi}{8}) + \sin(\frac{3\pi}{16}) + \sin(\frac{\pi}{4}) + \sin(\frac{5\pi}{16}) + \sin(\frac{3\pi}{8}) + \sin(\frac{7\pi}{16})) + \sin(\frac{\pi}{2})) = 0.9967852$$

$err_{absolue} = 0.0032$. Elle est environ 4 fois plus petite que l'erreur obtenue avec 4 intervalles.

3.3.3 Formule de Simpson 1/3 (simple):

$$\int_{x_0}^{x_2} f(x)dx \approx \frac{h}{3}(f(x_0) + 4f(x_1) + f(x_2))$$

Exemple : $\int_0^{\frac{\pi}{2}} \sin(x)dx$

$$I = \int_0^{\frac{\pi}{2}} \sin(x)dx \approx \frac{\frac{\pi}{4}}{3}(\sin(0) + 4\sin(\frac{\pi}{4}) + \sin(\frac{\pi}{2})) = 1.0022799$$

Résultat plus précis par rapport à la MTS mais il demeure peu satisfaisant.

3.3.4 Méthode de Simpson 1/3 composée:

On divise $[a,b]$ en $2n$ intervalles et on utilise n fois la méthode de Simpson 1/3 simple

$$h=\frac{b-a}{2n};\quad x_i=a+ih,\quad i=0,1...2n$$

$$\int_a^b f(x)dx=\sum_{i=0}^{n-1}\int_{x_{2i}}^{x_{2i+2}} f(x)dx\approx\sum_{i=0}^{n-1}\frac{h}{3}(f(x_{2i})+4f(x_{2i+1})+f(x_{2i+2}))$$

$$=\frac{h}{3}\begin{bmatrix}(f(x_0)+4f(x_1)+f(x_2)+\\ f(x_2)+4f(x_3)+f(x_4)...+\\ f(x_{2n-4})+4f(x_{2n-3})+f(x_{2n-2})+\\ f(x_{2n-2})+4f(x_{n-1})+f(x_{2n})\end{bmatrix}$$

$$=\frac{h}{3}\left[f(x_0)+4f(x_1)+2f(x_2)+4f(x_3)+2f(x_4)+...+4f(x_{2n-3})+2f(x_{2n-2})+4f(x_{n-1})+f(x_{2n})\right]$$

Tous les termes de rang impair sont multipliés par 4 tandis que ceux de rang pair sont multipliés par 2, sauf le premier et le dernier terme.

Exemple:

Évaluer $\int_0^{\frac{\pi}{2}}\sin(x)dx$ par la méthode des trapèzes composée en utilisant:

a) 4 intervalles. b) 8 intervalles.

$$a)\ h=\frac{\frac{\pi}{2}-0}{4}=\frac{\pi}{8}$$

$$I=\int_0^{\frac{\pi}{2}}\sin(x)dx\approx\frac{\frac{\pi}{8}}{3}(\sin(0)+4\times\sin(\frac{\pi}{8})+2\times\sin(\frac{\pi}{4})+4\times\sin(\frac{3\pi}{8})+\sin(\frac{\pi}{2}))=1.0001346$$

$$b)\ h=\frac{\frac{\pi}{2}-0}{8}=\frac{\pi}{16}$$

$$I=\int_0^{\frac{\pi}{2}}\sin(x)dx\approx\frac{\frac{\pi}{16}}{3}(\sin(0)+4\times\sin(\frac{\pi}{16})+2\times\sin(\frac{\pi}{8})+4\times\sin(\frac{3\pi}{16})+2\times\sin(\frac{\pi}{4})+4\times\sin(\frac{5\pi}{16})+$$
$$2\times\sin(\frac{3\pi}{8})+4\times\sin(\frac{7\pi}{16})+\sin(\frac{\pi}{2}))=1.000008296$$

En passant de 4 à 8 intervalles on divise l'erreur par un facteur d'environ 16.22

Exemple :

Calculer $\int_0^1 e^{-x^2}dx$ à l'aide de la méthode de Simpson 1/3 composée avec 8 intervalles.

$$h = \frac{1-0}{8} = \frac{1}{8}$$

$$I = \int_0^1 e^{-x^2} dx \approx \frac{\frac{1}{8}}{3}(e^0 + 4\times e^{-0.125^2} + 2\times e^{-0.25^2} + 4\times e^{-0.375^2} + 2\times e^{-0.5^2} + 4\times e^{-0.625^2} + 2\times e^{-0.75^2} +$$
$$4\times e^{-0.875^2} + e^{-1.0}) = 0.7468261205$$

Calcul de $\int_0^1 e^{-x^2} dx$

Méthode des trapèzes composée	0.746809163
Méthode de Simpson 1/3 composée	0.746824133
Solution exacte à 9 décimales	0.746824133

Ce qui démontre la supériorité de la méthode de Simpson.

3.3.5 Formule de Simpson 3/8:

$$\int_{x_0}^{x_3} f(x)dx \approx \frac{3h}{8}(f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3))$$

On peut composer cette méthode en divisant l'intervalle d'intégration $[a,b]$ en $3n$ sous-intervalles de longueur $h = \frac{b-a}{3n}$ et en utilisant la formule de Simpson 3/8 simple dans chaque triplet de sous-intervalle. On obtient alors :

$$\int_a^b f(x)dx = \sum_{i=0}^{n-1} \int_{x_{3i}}^{x_{3i+2}} f(x)dx \approx \sum_{i=0}^{n-1} \frac{3h}{8}(f(x_{3i}) + 3f(x_{3i+1}) + 3f(x_{3i+2}) + f(x_{3i+3}))$$
$$= \frac{3h}{8}\left[(f(x_0) + 3f(x_1) + 3f(x_2) + 2f(x_3) + 3f(x_4) + \ldots + 2f(x_{3n-3}) + 3f(x_{3n-2}) + 3f(x_{3n-1}) + f(x_{3n})\right]$$

Chapitre 4: Résolution d'équations différentielles

4.1 Introduction:

La résolution numérique des équations différentielles (ED) est probablement le domaine de l'analyse numérique où les applications sont les plus nombreuses. On aboutit souvent à la résolution d'équations différentielles, de systèmes d'équations différentielles, ou plus généralement d'équations aux dérivées partielles. Nous prenons comme point de départ la formulation générale d'une équation différentielle d'ordre 1 avec condition initiale. La tache consiste à déterminer une fonction $y(t)$ solution de : $\begin{cases} y'(t) = f(t, y(t)) \\ y(t_0) = y_0 \end{cases}$ (4.1)

Il s'agit d'obtenir $y(t)$ pour $t > 0$ si l'on cherche une solution analytique, ou une approximation de $y(t)$ si l'on utilise une méthode numérique.

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

Exemple:

Soit l'équation différentielle de 1er ordre : $\begin{cases} y'(t) = t \\ y(0) = 1 \end{cases}$ $y(t)$?

La solution générale est obtenue comme suit :

$$\int y'(t)dt = \int t\, dt \Rightarrow y(t) = \frac{t^2}{2} + c \qquad c : \text{est une constante.}$$

$$c\,? \quad on\ a\ y(0) = 1 = c \Rightarrow y(t) = \frac{t^2}{2} + 1 \quad \text{qui représente la solution particulière.}$$

4.2 Méthode d'Euler:

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

On cherche le point $((t_1, y_1) \approx (t_1, y(t_1))$ avec $t_1 = t_0 + h$

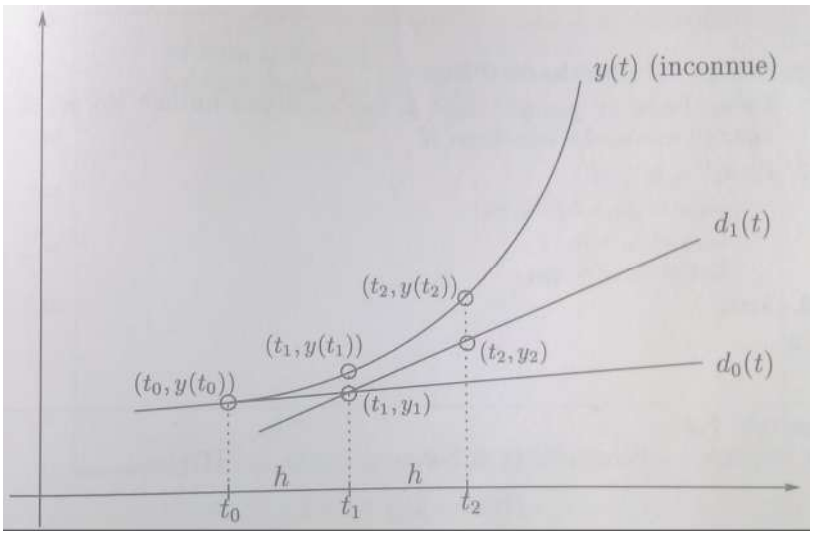

Figure 4.1

Nous n'avons pas l'équation de la courbe $y(t)$, mais nous en connaissons la pente $y'(t)$ en $t = t_0$.

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

La droite passant par (t_0, y_0) et de pente $f(t_0, y_0)$ a l'équation suivante : (voir Figure 4.1)

$$d_0(t) = y_0 + f(t_0, y_0)(t - t_0)$$

En $t = t_1$ on a:

$$d_0(t_1) = y_0 + f(t_0, y_0)(t_1 - t_0) = y_0 + h\, f(t_0, y_0) = y_1$$

$d_0(t_1)$ est proche d la solution analytique $y(t_1)$, c.à.d: $y(t_1) \approx y_1 = d_0(t_1) = y_0 + h\, f(t_0, y_0)$

Le plus souvent $y_1 = y(t_1)$

Si l'on souhaite faire une 2ième itération et obtenir une approximation de $y(t_2)$, on peut refaire l'analyse précédente à partir du point $(t_1, y(t_1))$ *en* $t = t_1$.En $t = t_1$, la pente de la solution analytique est $y'(t_1) = f(t_1, y(t_1))$

$y(t_1)$ est inconnu mais on connait y_1 (approximation de $y(t_1)$)

$$y'(t_1) = f(t_1, y(t_1)) \approx f(t_1, y_1)$$

L'équation de la droite $d_1(t)$ est :

$$d_1(t) = y_1 + f(t_1, y_1)(t - t_1)$$
$$y(t_2) \approx y_2 = d_1(t_2) = y_1 + h\, f(t_1, y_1)$$

L'erreur $|y(t_i) - y_i|$ augmente légèrement avec i.

On arrive à l'algorithme suivant:

Soit $[a,b]$ l'intervalle d'intégration. On subdivise $[a,b]$ en N intervalles de longueur

$$h = \frac{b-a}{N} \qquad \text{(le pas)}$$

$$\begin{cases} y_{n+1} = y_n + h\, f(t_n, y_n) \\ t_{n+1} = t_{n+1} + h \end{cases} \qquad 0 \le n \le N-1$$

Exemple:

Soit le problème de Cauchy

$$\begin{cases} y'(t) = y(t) + t \\ y(0) = 1 \end{cases} \qquad (4.2)$$

On veut approcher, à 10^{-3}, la solution de (4.2) en $t = 1$ à l'aide de la méthode d'Euler, en subdivisant $[0,1]$ en 10 parties égales.

$$h = \frac{1-0}{10} = 0.1$$

$$\begin{cases} y_{n+1} = y_n + h\, f(t_n, y_n) = y_n + h\,(y_n + t_n) \\ t_{n+1} = t_{n+1} + h \end{cases} \qquad 0 \le n \le 9$$

On calcule les valeurs du tableau suivant:

n	0	1	2	3	4	5	6	7	8	9	10
t_n	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
y_n	1	1.1	1.22	1.362	1.5282	1.7210	1.9431	2.1974	2.4871	2.8158	3.1874

On trouve $y(1) \approx 3.187.$ la solution exacte de l'équation (4.2) est donnée par: $y(t) = 2 \times \exp(t) - t - 1,$ ce qui donne $y(1) = 3.437.$ L'approximation calculée est très grossière.

Remarque: l'erreur dans la méthode d'Euler est relativement importante. Elle peut être améliorée en choisissant plus petit le pas, ce qui augmente considérablement le volume des calculs à effectuer. Elle est relativement peu utilisée en raison de sa faible précision.

Définition: méthode numérique à un pas:

Une méthode de résolution d'équation différentielle est dite à un pas si elle est de la forme:

$y_{n+1} = y_n + \phi(t_n, y_n)$ (équation aux différences)

Où ϕ: fonction quelconque.

La méthode est à un pas si, pour obtenir la solution en $t = t_{n+1}$, on doit utiliser la solution numérique au temps t_n seulement.

Pour Euler : $\phi(t, y) = f(t, y)$

4.3 Méthode de Taylor d'ordre 2:

On cherche, au temps $t = t_n$, une approximation de la solution en $t = t_{n+1}$. On a :

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

On a aussi: $y'(t) = f(t, y)$

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

En utilisant la règle :

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

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

On obtient:

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

En négligeant tous les termes d'ordre ≥ 3 on arrive à poser:

$$y(t_{n+1}) \approx y(t_n) + h\, f(t_n, y(t_n)) + \frac{h^2}{2}\left(\frac{\partial f(t_n, y(t_n))}{\partial t} + \frac{\partial f(t_n, y(t_n))}{\partial y} f(t_n, y(t_n)) \right) \qquad (4.3)$$

Algorithme de Taylor d'ordre 2:

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

2. *pour* $0 \leq n \leq N$

$$y_{n+1} = y_n + h\, f(t_n, y_n) + \frac{h^2}{2}\left(\frac{\partial f(t_n, y_n)}{\partial t} + \frac{\partial f(t_n, y_n)}{\partial y} f(t_n, y_n) \right)$$

$t_{n+1} = t_n + h$

écrire t_{n+1} *et* y_{n+1}

3. Arrêt

Exemple:

$$\begin{cases} y'(t) = -y(t) + t + 1 \\ y(0) = 1 \end{cases} \quad h = 0.1$$

On a $y_{10\_euler} = 1.348678$

Solution exacte:

$y_{exacte} = y(t_{10}) = 1.367879; \quad err_1 = \left| y(t_{10}) - y_{10\_euler} \right| = 0.01920$

$y_{10\_Taylor} =; \quad err_2 = \left| y(t_{10}) - y_{10\_Taylor} \right| = 0.0006$

on note que $err_2 < \quad err_1$

$f(t,y) = -y + t + 1 \quad ; \quad \frac{\partial f}{\partial t} = 1 \quad ; \quad \frac{\partial f}{\partial y} = -1$

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

Itération 1: $y_1 = 1 + 0.1\,(-1 + 0 + 1) + \frac{0.1^2}{2}\left(1 + (-1)(-1 + 0 + 1)\right) = 1.005$

Itération 2: $y_2 = 1.005 + 0.1\,(-1.005 + 0.1 + 1) + \frac{0.1^2}{2}\left(1 + (-1)(-1.005 + 0.1 + 1)\right) = 1.019025$

$\vdots$

Itération 10: $y_{10} = 1.368541$

4.4 Méthodes de Runge-Kutta d'ordre 2:

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

$$y(t_{n+1}) = y(t_n) + h\,f(t_n, y(t_n)) + \frac{h^2}{2}\left(\frac{\partial f(t_n, y(t_n))}{\partial t} + \frac{\partial f(t_n, y(t_n))}{\partial y} f(t_n, y(t_n))\right) + O(h^3) \quad (4.4)$$

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

$$y(t_{n+1}) = y(t_n) + a_1 h\,f(t_n, y(t_n)) + a_2 h\,f(t_n + a_3 h, y(t_n) + a_4 h) \quad (4.5)$$

Où l'on doit déterminer les paramètres a_1, a_2, a_3, et a_4 de telle sorte que les expressions aient toutes deux une erreur en $O(h^3)$. On ne trouve aucune dérivée partielle dans cette expression. Pour y arriver, on doit recourir au développement de Taylor en 2 variables autour du point $(t_n, y(t_n))$.

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

La relation (**) devient alors:

$$y(t_{n+1}) = y(t_n) + (a_1 + a_2)\,h\,f(t_n, y(t_n)) + a_2 a_3 h^2 \frac{\partial f(t_n, y(t_n))}{\partial t} + a_2 a_4 h^2 \frac{\partial f(t_n, y(t_n))}{\partial y} + O(h^3) \quad (4.6)$$

Par identification des expressions (4.4) et (4.6) terme à terme on obtient:

$$a_1 + a_2 = 1 \quad ; \quad a_2 a_3 = \frac{1}{2} \quad ; \quad \frac{f(t_n, y(t_n))}{2} = a_2 a_4$$

qui est un système (non linéaire) sous-déterminé en sens qu'il y a moins d'équations que d'inconnues et qu'il n'a donc pas de solution unique. Cela offre une marge de manœuvre qui favorise la mise au point de plusieurs variantes de la méthode de Runge-Kutta.

Variantes de la méthode de Runge-Kutta:

4.4.1 Méthode d'Euler modifiée:

$$a_1 = a_2 = \frac{1}{2} \quad ; \quad a_3 = 1 \quad et \quad a_4 = f(t_n, y(t_n))$$

Algorithme de la méthode d'Euler modifiée:

1. Etant donné un pas de temps h, une condition initiale (t_0, y_0) et un nombre maximal d'itérations N.
2. *pour* $0 \le n \le N$

$$\hat{y} = y_n + h\, f(t_n, y_n)$$

$$y_{n+1} = y_n + \frac{h}{2}\left(f(t_n, y_n) + f(t_n + h, \hat{y}) \right)$$

$$t_{n+1} = t_n + h$$

écrire t_{n+1} *et* y_{n+1}

3. Arrêt

Exemple:

$$\begin{cases} y'(t) = -y(t) + t + 1 \\ y(0) = 1 \end{cases} \quad h = 0.1$$

Itération 1:

$\hat{y} = 1 + 0.1\,(-1 + 0 + 1) = 1$ (résultat obtenu avec Euler)

$y_1 = 1 + 0.05\big((-1 + 0 + 1) + (-1 + 0.1 + 1)\big) = 1.005$

Itération 2:

$\hat{y} = 1.005 + 0.1\,(-1.005 + 0.1 + 1) = 1.0145$

$y_2 = 1.005 + 0.05\big((-1.005 + 0.1 + 1) + (-1.0145 + 0.1 + 1)\big) = 1.019025$

Une autre méthode de Runge-Kutta d'ordre 2 qui est très utilisée est la méthode du point milieu, qui correspond au choix suivant des coefficients a_i.

4.4.2 Méthode du point milieu:

$$a_1 = 0 \quad ; \quad a_2 = 1 \quad ; \quad a_3 = \frac{1}{2} \quad et \quad a_4 = \frac{f(t_n, y(t_n))}{2}$$

Algorithme de la méthode du point milieu:

1. Etant donné un pas de temps h, une condition initiale (t_0, y_0) et un nombre maximal

d'itérations N.

2. *pour* $0 \le n \le N$

$k_1 = h\ f(t_n, y_n)$

$y_{n+1} = y_n + h\left(f(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}) \right)$

$t_{n+1} = t_n + h$

écrire t_{n+1} *et* y_{n+1}

3. Arrêt

4.5 Méthode de Runge-Kutta d'ordre 4:

Algorithme:

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

2. *pour* $0 \le n \le N$

$k_1 = h\ f(t_n, y_n)$

$k_2 = h\ f(t_n + \frac{h}{2}, y_n + \frac{k_1}{2})$

$k_3 = h\ f(t_n + \frac{h}{2}, y_n + \frac{k_2}{2})$

$k_4 = h\ f(t_n + h, y_n + k_3)$

$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$

$t_{n+1} = t_n + h$

écrire t_{n+1} *et* y_{n+1}

3. Arrêt

Remarque:

La méthode de Runge-Kutta d'ordre 4 est très fréquemment utilisée en raison de sa grande précision qui est mise en évidence dans l'exemple suivant.

Exemple:

$$\begin{cases} y'(t) = -y(t) + t + 1 \\ y(0) = 1 \end{cases} \quad h = 0.1$$

Itération 1:

$k_1 = 0.1 f(0,1) = 0.1(-1+1) = 0$

$k_2 = 0.1 f(0+0.05, 1+0) = 0.1(-1+1.05) = 0.005$

$k_3 = 0.1f(0+0.05,1+1+0.0025) = 0.1(-1.0025+1.05) = 0.00475$

$k_4 = 0.1f(0+0.1,1+1+0.00475) = 0.1(-1.00475+1.1) = 0.009525$

$y_1 = 1+\frac{1}{6}\left(0+2(0.005)+2(0.0045)+0.009525\right) = 1.0048375$

Itération 2:

$k_1 = 0.1f(0.1,1.0048375) = 0.1(-1.0048375+0.+1) = 0.00951625$

$k_2 = 0.1f(0.15,1.00951625) = 0.1(-1.00951625+0.15+1) = 0.014040438$

$k_3 = 0.1f(0.15,1.011857719) = 0.1(-1.011857719+0.15+1) = 0.0138142281$

$k_4 = 0.1f(0.2,1.018651728) = 0.1(-1.018651728+0.2+1) = 0.0181348272$

$y_2 = 1.0048375+\frac{1}{6}(k_1+2k_2+2k_3+k_4) = 1.0187309014$

$\vdots$

$y_{10} = y(1.0) = 1.3678797744$

$y(t_{10}) = 1.3678794412$: solution exacte

err$=\left|y(t_{10})-y_{10}\right| = 0.333\times10^{-6}$

On constate que l'erreur se situe autour de 10^{-6}.

Exemple:

$$\begin{cases} y'(t) = -y(t)+t+1 \\ y(0) = 1 \end{cases}$$

On recourt à 3 méthodes de résolution : la méthode d'Euler avec pas $h = 0.025$, la méthode d'Euler modifiée avec $h = 0.05$, et la méthode de Runge-Kutta d'ordre 4 avec $h = 0.1$.

Le tableau suivant présente les résultats obtenus en $t = 1$ pour ces différents choix. La valeur exacte de la solution est $y(1.0) = 1.3678794412$.

Comparaison des différentes méthodes : $y'(t) = -y(t)+t+1$				
Méthode	h	Nombre de pas	Résultat	Erreur
Euler	0.025	40	1.36323237417	0.464×10^{-2}
Euler modifiée	0.05	20	1.36803862167	0.159×10^{-3}
Runge-Kutta	0.1	10	1.36787977441	0.333×10^{-6}

Les résultats sont éloquents. Même en prenant un pas de temps quatre fois petit, la méthode d'Euler reste très imprécise par rapport à celle de Runge-Kutta d'ordre 4. On peut porter le même jugement sur la méthode d'Euler modifiée. Il est donc préférable d'utiliser des méthodes d'ordre aussi élevé que possible.

Chapitre 5: Interpolation polynomiale

5.1 Formulation du problème:

A partir d'une fonction $f(x)$ connue seulement en $(n+1)$ points de la forme $((x_i, f(x_i))\ pour\ i = 0,1,...n),$ peut on construire une approximation de $f(x)$, et ce, pour tout x? les points $(x_i, f(x_i))$ sont appelés points de collocation ou points d'interpolation et peuvent provenir de données expérimentales ou d'une table. En d'autres termes, si l'on ne connait que les points d'interpolation d'une fonction, peut-on obtenir une approximation de $f(x)$ pour une valeur de x différente des x_i? (voir Figure 5.1)

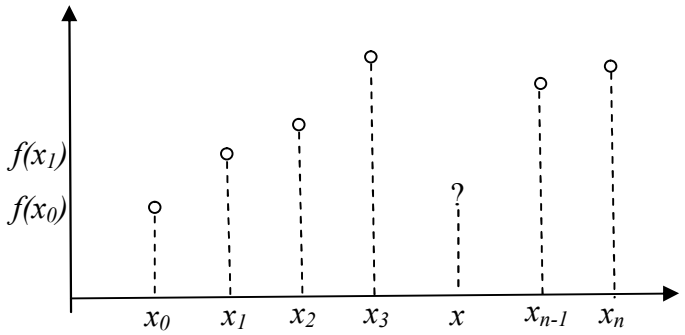

Figure 5.1: Problème d'interpolation

Théorème:

Un polynôme de degré n dont la forme générale est:

$$p_n(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + ... + a_nx^n \quad (5.1)$$

possède exactement n racines qui peuvent être réelles ou complexes conjuguées (on sait que r est une racine de $p_n(x)$ si $p_n(r) = 0$).

Corolaire: par $(n+1)$ points d'interpolation d'abscisses distinctes $((x_i, f(x_i)), i = 0,1,...n),$ on ne peut faire correspondre qu'un et un seul polynôme de degré n.

5.2 Matrice de Vandermonde:

Le problème d'interpolation consiste donc à déterminer l'unique polynôme de degré n passant par les $(n+1)$ points d'interpolation $(x_i, f(x_i))$. Une première tentative consiste à déterminer les inconnues a_i du polynôme (1) en vérifiant directement les $(n+1)$ équations de collocation:

$$p_n(x_i) = f(x_i) \quad pour\ i = 0,1,...n)$$
$$a_0 + a_1 x_i + a_2 x_i^{\ 2} + a_3 x_i^{\ 3} + ... + a_n x_i^{\ n} = f(x_i)$$

Qui est un système linéaire de $(n+1)$ équations à $(n+1)$ inconnues. Ce système s'écrit sous forme matricielle:

$$\underbrace{\begin{pmatrix} 1 & x_0 & x_0 & x_0 & \cdots & x_0 \\ 1 & x_1 & x_1^2 & x_1^3 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & x_2^3 & \cdots & x_2^n \\ 1 & x_3 & x_3^2 & x_3^3 & \cdots & x_3^n \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & x_n^3 & \cdots & x_n^n \end{pmatrix}}_{\text{Matrice de Vandermonde}} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \\ f(x_3) \\ \vdots \\ f(x_n) \end{pmatrix}$$

Exemple: calculer le polynôme passant par le points (0,1), (1,2), (2,9) et v(3,28).

On a 4 points, donc le polynôme est de degré 3.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 9 \\ 28 \end{pmatrix}$$

La solution (obtenue par la décomposition LU) est $\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T$. Donc $p_3(x) = 1 + x^3$

5.3 Interpolation de Lagrange:

L'interpolation de Lagrange est une façon simple et systématique de construire un polynôme d'interpolation. Etant donné $(n+1)$ points $((x_i, f(x_i)), i = 0,1,...n),$ on suppose un instant que l'on sait construire $(n+1)$ polynômes $L_i(x)$ de degré n et satisfaisant les conditions suivantes:

$$\begin{cases} L_i(x_i) = 1 & \forall i \\ L_i(x_j) = 0 & \forall j \neq i \end{cases}$$

La fonction (polynôme recherché) $L(x)$ est définie par : $L(x) = \sum_{i=0}^{n} f(x_i) L_i(x_i)$

Polynôme de degré 1:

Il s'agit de déterminer le polynôme de degré 1 dont la courbe (une droite) passe par les 2 points $(x_0, f(x_0))$ et $(x_1, f(x_1))$ On doit donc construire 2 polynômes $L_0(x)$ et $L_1(x)$ de degré 1 qui vérifient:

$$\begin{cases} L_0(x_0)=1 \\ L_i(x_1)=0 \end{cases} \quad \begin{cases} L_1(x_0)=0 \\ L_1(x_1)=1 \end{cases} \Rightarrow L_0(x)=\frac{(x-x_1)}{(x_0-x_1)}\,;\, L_1(x)=\frac{(x-x_0)}{(x_1-x_0)}$$

$$p_1(x)=f(x_0)L_0(x)+f(x_1)L_1(x)$$

Exemple: l'équation de la droite passant par les points (2,3) et (5,-6) est :

$$3\frac{(x-5)}{(2-5)}+(-6)\frac{(x-2)}{(5-2)}=-(x-5)-2(x-2)=-3x+9$$

Polynômes de degré 2:

On cherche le polynôme de degré 2 passant par les points $(x_0, f(x_0))$, $(x_1, f(x_1))$ et $(x_2, f(x_2))$.

On doit construire 3 fonctions $L_i(x)$:

$$L_0(x)=\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}\,;\, L_1(x)=\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}\,;\, L_2(x)=\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}$$

$$p_2(x)=f(x_0)L_0(x)+f(x_1)L_1(x)+f(x_2)L_2(x)$$

Polynômes de degré *n*:

Théorème: étant donné $(n+1)$ points d'interpolation $((x_i, f(x_i)), i=0,1,...n)$, l'unique polynôme d'interpolation de degré n passant par tous ces points peut s'écrire: $p_n(x)=\sum_{i=0}^{n} f(x_i)L_i(x)$

Où les $(n+1)$ fonctions $L_i(x)$ sont définies par la formule de Lagrange:

$$L_i(x)=\frac{(x-x_0)(x-x_1)...(x-x_{i-1})(x-x_{i+1})...(x-x_n)}{(x_i-x_0)(x_i-x_1)...(x_i-x_{i-1})(x_i-x_{i+1})...(x_i-x_n)}$$

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

$$L_0(x)=\frac{(x-x_1)(x-x_2)...(x-x_n)}{(x_0-x_1)(x_0-x_2)...(x_0-x_n)}$$

$$L_1(x)=\frac{(x-x_0)(x-x_2)...(x-x_n)}{(x_1-x_0)(x_1-x_2)...(x_0-x_n)}$$

Exemple:

Déterminer l'interpolation de Lagrange pour les points (0,1), (1,2), (2,9) et (3,28)

$$p_3(x)=1\frac{(x-1)(x-2)(x-3)}{(x_0-1)(x_0-2)(x_0-3)}+2\frac{(x-0)(x-2)(x-3)}{(x_1-0)(x_1-2)(x_1-3)}+9\frac{(x-0)(x-1)(x-2)}{(x_2-0)(x_2-1)(x_2-2)}$$

$$p_3(x)=x^3+1$$

5.4 Polynôme de Newton:

Définition: on définit les premières différences divisées de la fonction $f(x)$ par:

$$f[x_i, x_{i+1}] = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$$

Les deuxièmes différences divisées de la fonction f(x) sont définies à partir des premières différences divisées par la relation:

$$f[x_i, x_{i+1}, x_{i+2}] = \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{(x_{i+2} - x_i)}$$

De même, les nièmes différences divisées de la fonction $f(x)$ sont définies à partir des (n-1)ièmes différences divisées de la façon suivante:

$$f[x_0, x_1, x_2, \ldots x_n] = \frac{f[x_1, x_2, \ldots x_n] - f[x_0, x_1, x_2, \ldots x_{n-1}]}{(x_n - x_0)}$$

Théorème: l'unique polynôme de degré n passant par les $(n+1)$ points d'interpolation $((x_i, f(x_i)), i = 0,1,\ldots n),$ peut s'écrire selon la formule d'interpolation de Newton:

$$p_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + a_3(x - x_0)(x - x_1)(x - x_2) \ldots + a_{n-1}(x - x_0)(x - x_1) \ldots (x - x_{n-2}) + a_n(x - x_0)(x - x_1) \ldots (x - x_{n-1}) \qquad (5.2)$$

Ou encore sous la forme récursive:

$$p_n(x) = p_{n-1}(x) + a_n(x - x_0)(x - x_1) \ldots (x - x_{n-1})$$

Les coefficients de ce polynôme sont les différences divisées:

$$a_i = f[x_0, x_1, x_2, \ldots x_n] \quad pour\ 0 \le i \le n$$

$$a_0 = f(x_0) \ ; \ a_1 = f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

$$a_2 = f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_2, x_1]}{(x_2 - x_0)}$$

$$a_3 = f[x_0, x_1, x_2, x_3]$$

Exemple: les polynômes de Newton de degré 1 et 2 sont comme suit:

$$p_1(x) = f(x_0) + f[x_0, x_1](x - x_0)$$

$$p_2(x) = \underbrace{f(x_0) + f[x_0, x_1](x - x_0)}_{p_1(x)} + f[x_0, x_1, x_2](x - x_0)(x - x_1)$$

$$p_2(x) = p_1(x) + f[x_0, x_1, x_2](x - x_0)(x - x_1)$$

Remarque: une fois les coefficients a_i connus, on peut évaluer le polynôme de Newton. On écrit le polynôme (5.2) sous la forme:

La manière la plus simple consiste à construire une table dite des différences divisées de la façon suivante:

Table des différences divisées			
$f(x_i)$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$f[x_i, x_{i+1}, x_{i+2}]$
$f(x_0)$			
	$f[x_0, x_1]$		
$f(x_1)$		$f[x_0, x_1, x_2]$	
	$f[x_1, x_2]$		$f[x_0, x_1, x_2, x_3]$
$f(x_2)$		$f[x_1, x_2, x_3]$	
	$f[x_2, x_3]$		
$f(x_3)$			

La construction de cette table est simple. Nous nous sommes arrêtés au 3ièmes différences divisées, mais les autres s'obtiendraient de la même manière. Les 1ères différences divisées découlent de la définition. Par la suite, pour obtenir par exemple $f[x_0, x_1, x_2]$, il suffit de soustraire les deux termes adjacents $f[x_1, x_2] - f[x_0, x_1]$ et diviser le résultat par $(x_2 - x_0)$

$$f[x_1, x_2] = \frac{f(x_2) - f(x_1)}{x_2 - x_1}$$

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$$

$$f[x_0, x_1, x_2, x_3] = \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0}$$

La formule de Newton utilise la diagonale principale de cette table.

Exemple: construire la table des différences divisées des points (0,1), (1,2), (2,9) et (3,28)

x_i	$f(x_i)$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$f[x_i, x_{i+1}, x_{i+2}, x_{i+3}]$
0	1			
		1		
1	2		3	
		7		1
2	9		6	
		19		
3	28			

Suivant la formule de Newton (5.2) avec $x_0 = 0$, le polynôme d'interpolation est:

$$p_3(x) = 1 + 1(x-0) + 3(x-0)(x-1) + 1(x-0)(x-1)(x-2) = x^3 + 1$$

qui est le même polynôme (en vertu de l'unicité) que celui obtenu par la méthode de Lagrange. On remarque de plus que $p_2(x) = 1 + 1(x-0) + 3(x-0)(x-1)$ passe quant à lui par les 3 premiers points d'interpolation.

Si l'on souhaite ajouter un point de collocation et calculer un polynôme de degré 4, il n'est pas nécessaire de tout recommencer. Par exemple, si l'on veut inclure le point (5,54), on peut compléter la table de différences divisées déjà utilisée.

x_i	$f(x_i)$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$f[x_i, x_{i+1}, x_{i+2}, x_{i+3}]$	$f[x_i, x_{i+1}, ..., x_{i+4}]$
0	1				
		1			
1	2		3		
		7		1	
2	9		6		–3 / 5
		19		–2	
3	28		–2		
		13			
5	54				

Ce polynôme de degré 4 est alors: $p_4(x) = p_3(x) - \frac{3}{5}(x-0)(x-1)(x-2)(x-3)$

qui est tout simplement le polynôme de degré 3 déjà calculé auquel on a ajouté une correction de degré 4.

Erreur d'interpolation:

L'interpolation permet, à partir d'un certain nombre de données sur les valeurs d'une fonction, de faire l'approximation de $f(x)$ en tout point x. Toutefois cette opération entraîne une erreur d'interpolation. On peut exprimer cette erreur comme suit:

$$f(x) = p_n(x) + E_n(x)$$
$$E_n(x) = f(x) - p_n(x)$$

$E_n(x_i) = 0 \quad pour \;\; i = 0,...n$ car le polynôme passe exactement par ces points.

Théorème: soit $x_0 < x_1 < ... < x_n$, les abscisses des points de collocation. On suppose que la fonction $f(x)$ est définie dans $[x_0, x_n]$ et qu'elle est $(n+1)$ fois dérivable dans $]x_0, x_n[$. Alors pour tout x compris dans $[x_0, x_n]$, il existe $\varepsilon(x) \in]x_0, x_n[$ tel que:

$$E_n(x) = \frac{f^{(n+1)}(\varepsilon(x))}{(n+1)!}(x-x_0)(x-x_1)...(x-x_n)$$

qui représente l'expression analytique de l'erreur d'interpolation.

5.5 Splines cubiques:

L'utilisation de polynômes de degré élevé est parfois délicate et peut mener à des erreurs d'interpolation importantes. De plus, il est parfois nécessaire d'obtenir des courbes très régulières passant par un grand nombre de points. Le problème lorsqu'on utilise des polynômes de degré faible, provient du fait qu'il faut en utiliser plusieurs pour relier touts les points. C'est la cas de l'interpolation linéaire par morceaux.

Une vois très populaire consiste à utiliser dans chaque intervalle $[x_{i-1}, x_i]$ un polynôme de degré 3 de la forme : $p_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i \quad pour \ \ i = 1,2,...n$, et à relier ces différents polynômes de façon à ce que la courbe résultante soit 2 fois différentiable. C'est l'interpolation par **Splines cubiques** (voir Figure 5.2).

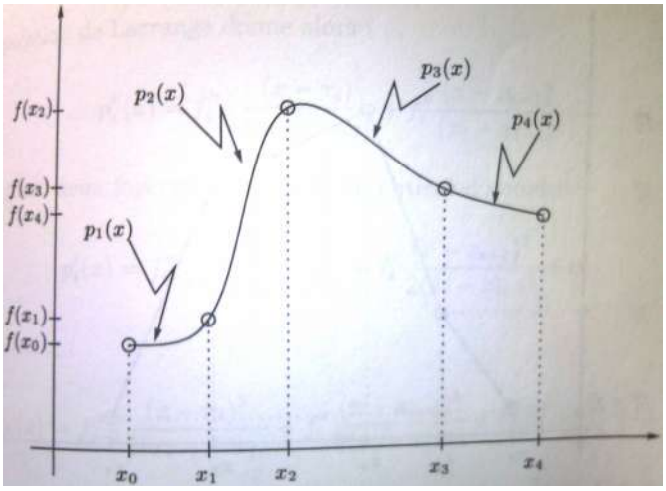

Figure 5.2 : Splines cubiques : *n* polynômes de degrés 3

Supposons que l'on ait $(n+1)$ points d'interpolation $\{(x_i, f(x_i)), i = 0,1,...n\}$, donc n intervalles $[x_0, x_i] \Rightarrow$ il y a $4n$ coefficients $(a_i, b_i, c_i, d_i, \ i = 1,2,...n)$

Les contraintes imposées aux n polynômes de degré 3 sont:

$$p_1(x_0) = f(x_0) \quad (p_1(x) \text{ passe par la première extrémité})$$

$$p_n(x_n) = f(x_n) \quad (p_n(x) \text{ passe par la dernière extrémité})$$

$$\begin{aligned} &p_i(x_i) = f(x_i) \quad i = 1,2,...n-1 \\ &p_{i+1}(x_i) = f(x_i) \quad i = 1,2,...n-1 \\ &p_i'(x_i) = p_{i+1}'(x_i) \quad i = 1,2,...n-1 \\ &p_i''(x_i) = p_{i+1}''(x_i) = f_i'' \quad i = 1,2,...n-1 \end{aligned} \qquad (5.3)$$

On a donc $2 + 2(n-1) + 2(n-1) = 4n - 2$ conditions.

Or, il faut trouver sur chaque intervalle 4 coefficients, il manque 2 conditions.

L'équation de la Spline dans l'intervalle $[x_{i-1}, x_i]$ est donnée par:

$$p_i(x) = -f''_{i-1}\frac{(x-x_i)^3}{6h_i} + f''_i\frac{(x-x_{i-1})^3}{6h_i} - \left(\frac{f(x_{i-1})}{h_i} - \frac{h_i f''_{i-1}}{6}\right)(x-x_i) + \left(\frac{f(x_i)}{h_i} - \frac{h_i f''_i}{6}\right)(x-x_{i-1}) \quad (5.4)$$

Il reste à déterminer les f''_i. Ces dernières sont déterminées à partir de l'équation (5.3). Après simplification, on obtient:

$$\frac{h_i}{(h_i+h_{i+1})}f''_{i-1} + 2f''_i + \frac{h_{i+1}}{(h_i+h_{i+1})}f''_{i+1} = 6f[x_{i-1}, x_i, x_{i+1}] \quad pour \;\; i = 1,2,...n-1 \quad (5.5)$$

avec $h_i = x_i - x_{i-1}\,; i = 1,...n$

Si on pose $f''_0 = 0$ *et* $f''_n = 0$ alors la Spline est complètement déterminée et s'appelle Spline naturelle. On peut aussi imposer : $f''_0 = f''_1 \;\; (f''_0 - f''_1 = 0)$
$f''_0 = f''_1 \;\; (f''_0 - f''_1 = 0)$

Remarque : pour effectuer une interpolation à l'aide des Splines cubiques, il faut en premier lieu résoudre le système (5.5) complété par les conditions aux extrémités. Par la suite, on doit déterminer l'intervalle dans lequel se situe le point d'interpolation x et calculer le polynôme dans cet intervalle en utilisant l'équation de la Spline.

Exemple: calculer la Spline cubique pour les 4 points suivants : (1,1), (2,4), (4,9), (5,11).

x_i	$f(x_i)$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	h
0	1			
		3		
1	2		$-1/6$	1
		$5/2$		
2	4		$-1/6$	2
		2		
3	5			1

$$\frac{h_i}{(h_i+h_{i+1})}f''_{i-1} + 2f''_i + \frac{h_{i+1}}{(h_i+h_{i+1})}f''_{i+1} = 6f[x_{i-1}, x_i, x_{i+1}] \quad pour \;\; i = 1,2,...n-1$$

avec $h_i = x_i - x_{i-1}\,; i = 1,...n$

Itération (1) du système (5.5) (i=1)

$$\frac{1}{3}f''_0 + 2f''_1 + \frac{2}{3}f''_2 = 6\left(-\frac{1}{6}\right)$$

Itération(2) : i=2

$$\frac{2}{3}f_1^{"}+2f_2^{"}+\frac{1}{3}f_3^{"}=6\left(-\frac{1}{6}\right)$$

Pour obtenir la Spline naturelle $(f_0^{"}=f_3^{"}=0)$, on résout le système:

$$\begin{pmatrix}1 & 0 & 0 & 0\\ \frac{1}{3} & 2 & \frac{2}{3} & 0\\ 0 & \frac{2}{3} & 2 & \frac{1}{3}\\ 0 & 0 & 0 & 1\end{pmatrix}\begin{pmatrix}f_0^{"}\\ f_1^{"}\\ f_2^{"}\\ f_0^{"}\end{pmatrix}=\begin{pmatrix}0\\ -1\\ -1\\ 0\end{pmatrix}$$

Dont la solution est $f_0^{"}=0$; $f_1^{"}=-0.375$; $f_2^{"}=-0.375$ et $f_3^{"}$=0

L'équation de la Spline dans le 1er intervalle est alors:

$$p_1(x)=0\frac{(x-2)^3}{6}-0.375\frac{(x-1)^3}{6}-\left(\frac{1}{1}-\frac{0}{6}\right)(x-2)+\left(\frac{4}{1}-\frac{(1)(-0.375)}{6}\right)(x-1)$$

Que l'on peut simplifier en:

$$p_1(x)=-0.0625(x-1)^3-(x-2)+4.0625(x-1)$$

Ce polynôme n'est défini que dans l'intervalle $[1,2]$, on peut par exemple l'évaluer en $x=1.5$ pour obtenir 2.523475.

Si on cherche la valeur de la Spline en $x=3$, qui est située dans le 2ième intervalle $[2,4]$, on devrait obtenir l'équation de la Spline dans cet intervalle en posant $i=3$ dans l'équation de la Spline. On a alors :

$$p_2(x)=0.375\frac{(x-4)^3}{12}-0.375\frac{(x-2)^3}{12}-\left(\frac{4}{2}-\frac{(2)(-0.375)}{6}\right)(x-4)+\left(\frac{9}{2}-\frac{(2)(-0.375)}{6}\right)(x-2)$$

$$p_2(x)=0.03125(x-4)^3-0.03125(x-2)^3-2.125(x-4)+4.625(x-2)$$

La valeur de la Spline en $x=3$ est 6.6875. La Spline complète est illustrée sur la figure suivante:

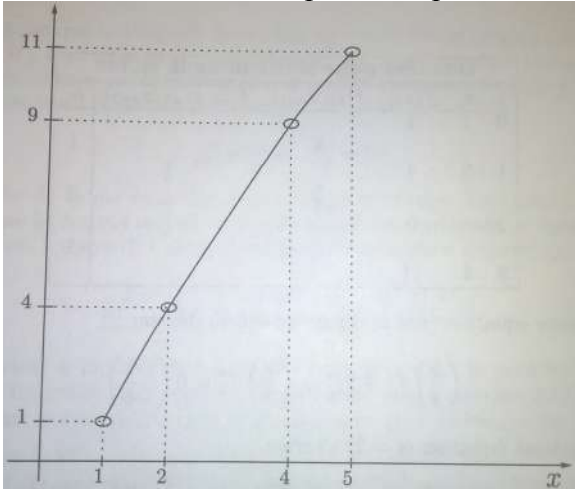

Figure 5.3: Spline passant par 4 points.

Chapitre 6: Approximation au sens des moindres carrées

6.1 Droite des moindres carrées:

L'objectif de cette méthode numérique est de déterminer une formule $y = f(x)$ qui relie un ensemble de points de données $(x_1, y_1),...(x_N, y_N)$ (issus d'une expérience par exemple), où les abscisses $\{x_k\}$ sont distinctes. On utilise des fonctions linéaires de la forme:

$$y = f(x) = Ax + B \qquad (6.1)$$

La valeur exacte $f(x_k)$ satisfit : $\begin{aligned} & f(x_k) = y_k + e_k \\ & \Rightarrow e_k = f(x_k) - y_k \quad pour \ 1 \le k \le N \end{aligned}$

e_k s'appelle erreur de mesure.

Pour déterminer la meilleure approximation linéaire de la forme (1) qui passe prés des points, on utilise la norme de l'erreur quadratique moyenne pour mesurer l'approximité entre la courbe $y = f(x)$ et les points de données.

Erreur quadratique moyenne (Eqm) : Root square error

$$E(f) = \left(\frac{1}{N} \sum_{k=1}^{N} \left| f(x_k) - y_k \right|^2 \right)^{\frac{1}{2}} \qquad (6.2)$$

Détermination de la droite des moindres carrées:

Soit $\{(x_k, y_k)\}_{k=1}^{N}$ un ensemble de N points, où les abscisses $\{x_k\}$ sont distinctes. La droite des moindres carrées (DMC) $y = f(x) = Ax + B$ est la droite qui minimise l'erreur quadratique moyenne. La quantité $E(f)$ sera minimale si et seulement si la quantité $NE(f)^2 = \sum_{k=1}^{N} \left(Ax_k + B - y_k \right)^2$ est minimale. Cette dernière peut être interprétée géométriquement par la minimisation de la somme des carrées des distances verticales qui commencent aux points et se dirigent vers la ligne (voir Figure 6.1).

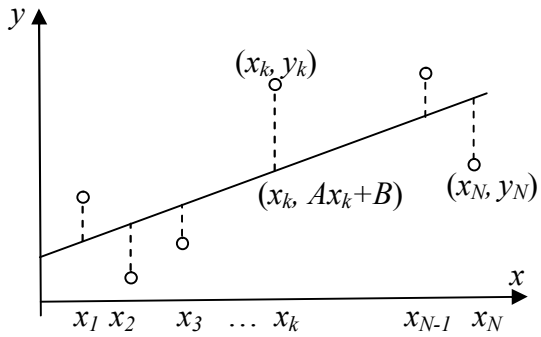

Figure 6.1: Distances verticales entre les points $\{(x_k, y_k)\}$ et la droite des moindres carrées $y=Ax+B$.

Théorème: (Droite des moindres carrées)

On suppose qu'on a N points $\{(x_k, y_k)\}_{k=1}^{N}$ où les abscisses sont distinctes. Les coefficients de la droite des moindres carrées $y = Ax + B$ sont la solution du système linéaire suivant:

$$\begin{cases} (\sum_{k=1}^{N} x_k^2)A + (\sum_{k=1}^{N} x_k)B = \sum_{k=1}^{N} x_k y_k \\ (\sum_{k=1}^{N} x_k)A + NB = \sum_{k=1}^{N} y_k \end{cases} \qquad (6.3)$$

Démonstration du théorème:

Géométriquement, on commence par la droite $y = Ax + B$. La distance verticale d_k qui sépare le point de départ (x_k, y_k) et le point final $(x_k, Ax_k + B)$ sur la droite est $d_k = | Ax_k + B - y_k |$ (voir Figure 1). On doit minimiser la somme des carrées des distances verticales d_k :

$$E(A,B) = \sum_{k=1}^{N} \left(Ax_k + B - y_k \right)^2 = \sum_{k=1}^{N} {d_k}^2$$

La valeur minimale de $E(A,B)$ est déterminée en posant : $\frac{\partial E}{\partial A} = 0$ et $\frac{\partial E}{\partial B} = 0$

$$\frac{\partial E(A,B)}{\partial A} = \sum_{k=1}^{N} 2\left(Ax_k + B - y_k \right)(x_k) = 2\sum_{k=1}^{N} \left(Ax_k^2 + Bx_k - x_k y_k \right)$$

$$\frac{\partial E(A,B)}{\partial B} = \sum_{k=1}^{N} 2\left(Ax_k + B - y_k \right) = 2\sum_{k=1}^{N} \left(Ax_k + B - y_k \right)$$

En mettant $\frac{\partial E(A,B)}{\partial A}$ et $\frac{\partial E(A,B)}{\partial B}$ et en utilisant les propriétés de distribution de la somme on obtient:

$$\begin{cases} \sum_{k=1}^{N}\left(Ax_k^2+Bx_k-x_k y_k\right) = A\sum_{k=1}^{N}x_k^2 + B\sum_{k=1}^{N}x_k - \sum_{k=1}^{N}x_k y_k = 0 \\ \sum_{k=1}^{N}\left(Ax_k+B-y_k\right) = A\sum_{k=1}^{N}x_k + NB - \sum_{k=1}^{N}y_k = 0 \end{cases}$$

Qui sont les mêmes équations du système (6.3).

Exemple:

Déterminer la droite des moindres carrées pour les points suivants : (-1,10), (0,9), (1,7), (2,5), (3,4), (4,3), (5,0), (6,-1).

	x_k	y_k	x_k^2	$x_k y_k$
	-1	10	1	-10
	0	9	0	0
	1	7	1	7
	2	5	4	10
	3	4	9	12
	4	3	16	12
	5	0	25	0
	6	-1	36	-6
Somme	20	37	92	25

$$\Rightarrow \begin{cases} 92A+20B=25 \\ 20A+8B=37 \end{cases} \Rightarrow \begin{matrix} A=1.6071429 \\ B=8.6428571 \end{matrix}$$

Donc $y=1.6071429x+8.6428571$

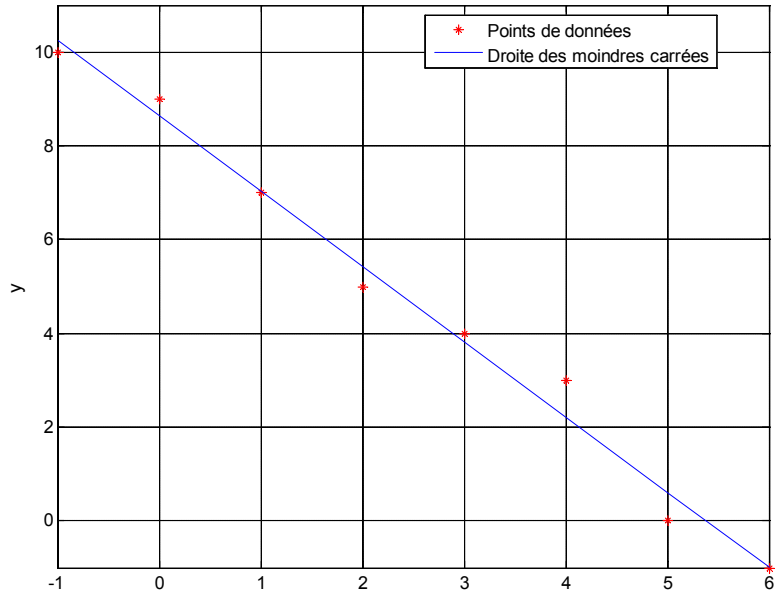

Figure 6.2: Droite des moindres carrées.

6.2 Parabole des moindres carrées:

Théorème:

Soit $\{(x_k, y_k)\}_{k=1}^{N}$ N points où les abscisses sont distinctes. Les coefficients de la parabole des moindres carrées $y = Ax^2 + Bx + C$ sont les solutions A, B et C du système linéaire:

$$\begin{cases} (\sum_{k=1}^{N} x_k^4)A + (\sum_{k=1}^{N} x_k^3)B + (\sum_{k=1}^{N} x_k^2)C = \sum_{k=1}^{N} y_k x_k^2 \\ (\sum_{k=1}^{N} x_k^3)A + (\sum_{k=1}^{N} x_k^2)B + (\sum_{k=1}^{N} x_k)C = \sum_{k=1}^{N} y_k x_k \\ (\sum_{k=1}^{N} x_k^2)A + (\sum_{k=1}^{N} x_k)B + NC = \sum_{k=1}^{N} y_k \end{cases} \qquad (6.4)$$

Démonstration du théorème:

Les coefficients A, B et C qui minimisent la quantité $E(A,B,C) = \sum_{k=1}^{N} \left(Ax_k^2 + Bx_k + C - y_k \right)^2$ sont obtenus en posant: $\frac{\partial E}{\partial A} = 0, \frac{\partial E}{\partial B} = 0, \frac{\partial E}{\partial C} = 0$

$$\frac{\partial E(A,B,C)}{\partial A} = 2\sum_{k=1}^{N} \left(Ax_k^2 + Bx_k + C - y_k \right)(x_k^2) = 0$$

$$\frac{\partial E(A,B,C)}{\partial B} = 2\sum_{k=1}^{N} \left(Ax_k^2 + Bx_k + C - y_k \right)(x_k) = 0$$

$$\frac{\partial E(A,B,C)}{\partial C} = 2\sum_{k=1}^{N} \left(Ax_k^2 + Bx_k + C - y_k \right)(1) = 0$$

Après simplification, on obtient les équations du système (6.4)

Exemple:

Déterminer la parabole des moindres carrées pour les points (-3,3), (0,1), (2,1) et (4,3).

Le système linéaire (6.4) pour déterminer A, B et C devient:

$$\begin{cases} 353A + 45B + 29C = 79 \\ 45A + 29B + 3C = 5 \\ 29A + 3B + 4C = 8 \end{cases}$$

La solution de ce système est :

$$\begin{cases} A = \dfrac{585}{3278} \\ B = \dfrac{-631}{3278} \\ C = \dfrac{1394}{1639} \end{cases}$$

$$\Rightarrow y = \frac{585}{3278}x^2 - \frac{631}{3278}x + \frac{1394}{1639}$$

$$y = 0.178462x^2 - 0.192495x + 0.850519$$

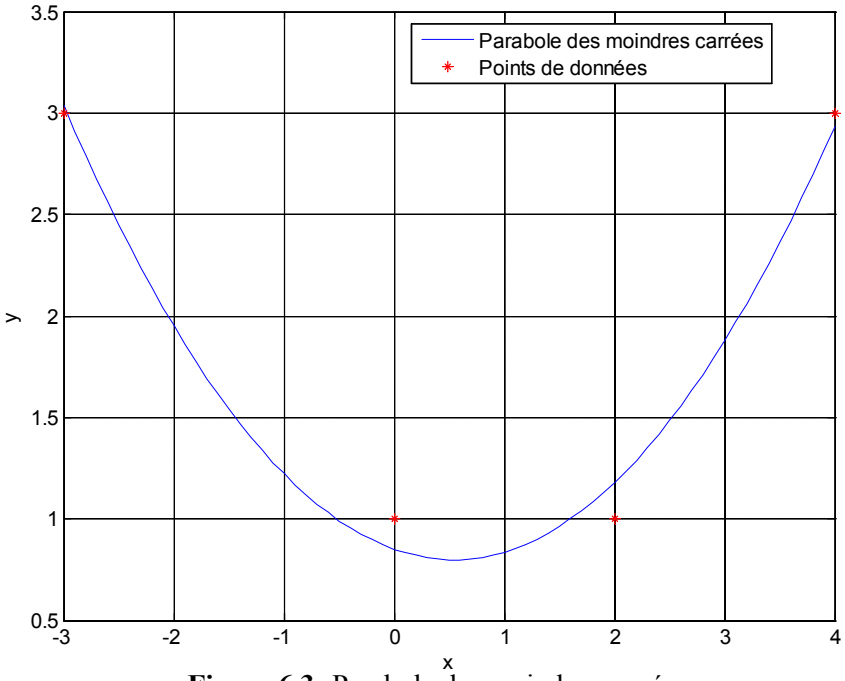

Figure 6.3: Parabole des moindres carrées.

Bibliographie

FORTIN, André. *Analyse numérique pour ingénieurs*. Presses inter Polytechnique, 2001.

QUARTERONI, Alfio, SACCO, Riccardo, et SALERI, Fausto. *Méthodes numériques pour le calcul scientifique*. Springer-France, 2000.

LAKRIB, Mustapha. *Cours d'analyse numérique.* 5ième édition. Office des publications universitaires, 2009.

GERALD Curtis F., Wheatley Patrick O. *Applied numerical analysis*. Addison wesley publishing company ; 5 edition 1994.

BURDEN Richard L. and Faires J. Douglas. *Numerical Analysis*. Ninth Edition, Brooks/ Cole , Cengage Learning, 2011.

SIAUW, Timmy et BAYEN, Alexandre. *An Introduction to MATLAB® Programming and Numerical Methods for Engineers*. Academic Press, 2014.

HAMMING, Richard. *Numerical methods for scientists and engineers*. Courier Corporation, 2012.

CHAPRA, Steven C. et CANALE, Raymond P. *Numerical methods for engineers*. New York : McGraw-Hill, 1998.

ISAACSON, Eugene et KELLER, Herbert Bishop. *Analysis of numerical methods*. Courier Corporation, 1994.

ATKINSON, Kendall et HAN, Weimin. *Theoretical numerical analysis*. Berlin : Springer, 2005.

HILDEBRAND, Francis Begnaud. *Introduction to numerical analysis*. Courier Corporation, 1987.

RALSTON, Anthony et RABINOWITZ, Philip. *A first course in numerical analysis*. Courier Corporation, 2001.