

Table des matières

Introduction	4
1 La résolution des équations non linéaires	5
1.1 Introduction	5
1.2 Préliminaires : séparations des racines	5
1.3 Méthode de dichotomie	6
1.3.1 Exercice d'application de la méthode de dichotomie	7
1.3.2 Programme en Matlab de la méthode de dichotomie	8
1.4 Méthode de Newton	8
1.4.1 Exercice d'application de la méthode de Newton	9
1.4.2 Programme en Matlab de la méthode de Newton	10
1.5 Méthode de la sécante	10
1.5.1 Exercice d'application de la méthode de la sécante	11
1.5.2 Programme en Matlab de la méthode de la sécante	11
1.6 Méthode du point fixe (des approximations successives)	12
1.6.1 Exercice d'application de la méthode du point fixe	13
1.6.2 Programme en Matlab de la méthode du point fixe	14
1.7 Série des exercices	14
2 Interpolation polynomiale	16
2.1 Introduction	16
2.2 Méthode d'interpolation de Lagrange	16
2.2.1 Existence et unicité du polynôme d'interpolation	16
2.2.2 Détermination pratique des polynômes de Lagrange	18
2.2.3 Exercices d'application de la méthode de l'interpolation de Lagrange	18
2.2.4 Programme en Matlab de la méthode de l'interpolation de Lagrange	19
2.3 Méthode d'interpolation de Newton	20
2.3.1 Différence divisée	20
2.3.2 Cas où les points d'interpolation sont équidistants	21
2.3.3 Exercice d'application de la méthode de l'interpolation de Newton	23
2.3.4 Programme en Matlab de la méthode de l'interpolation de Newton	23
2.4 Formule d'erreur d'interpolation	24
2.5 Série des exercices	25

3	Méthodes des intégrations numériques	27
3.1	Introduction	27
3.2	Méthode des trapèzes	27
3.2.1	Principe de la méthode	27
3.2.2	Formule généralisée des trapèzes	28
3.2.3	L'erreur de la méthode des trapèzes	29
3.2.4	Exercice d'application de la méthode des trapèzes	30
3.2.5	Programme en Matlab de la méthode des trapèzes	30
3.3	Méthode de Simpson	31
3.3.1	Principe de la méthode	31
3.3.2	Formule généralisée de Simpson	32
3.3.3	L'erreur de la méthode de Simpson	33
3.3.4	Exercice d'application de la méthode de Simpson	35
3.3.5	Programme en Matlab de la méthode de Simpson	36
3.4	Méthode de Newton	36
3.4.1	Principe de la méthode	36
3.4.2	Formule généralisée de Newton	37
3.5	Méthode de Newton-Cotes	37
3.5.1	Exercice d'application de la méthode de Newton	39
3.5.2	Programme en Matlab de la méthode de Newton	40
3.6	Méthode de Gauss	40
3.6.1	Comment calculer l'intégrale	41
3.6.2	Généralisation pour un intervalle fermé	44
3.6.3	Les différents types de la méthode de Gauss	44
3.7	Majoration de l'erreur de Gauss	46
3.7.1	Exercice d'application de la méthode de Gauss	47
3.7.2	Programme en Matlab de la méthode de Gauss	48
3.8	Série des exercices	48
4	La résolution du système des équations linéaires	50
4.1	Introduction	50
4.2	Généralités sur les matrices	50
4.2.1	Matrice inverse	50
4.2.2	Matrice diagonales et triangulaires	50
4.2.3	Matrice définie positive	51
4.2.4	Matrice de permutation	52
4.2.5	Les propriétés de déterminant	52
4.3	Généralités sur les systèmes linéaires	52
4.3.1	Cas d'une matrice diagonale	53
4.3.2	Cas d'une matrice triangulaire supérieure (où inférieure)	53
4.4	Les méthodes directes	54
4.4.1	Méthode de Gauss	54
4.4.2	Exercice d'application de la méthode de Gauss	56
4.4.3	Programme en Matlab de la méthode de Gauss	58
4.4.4	Factorisation LU	58
4.4.5	Exercice d'application de la méthode de Factorisation LU	59

4.4.6	Programme en Matlab de la méthode de Factorisation LU	60
4.4.7	Factorisation de Cholesky	60
4.4.8	Exercice d'application de la méthode de Factorisation de Cholesky . .	61
4.4.9	Programme en Matlab de la méthode de Factorisation de Cholesky .	62
4.4.10	Méthode de Gauss-Jordin	62
4.4.11	Exercice d'application de la méthode de Gauss-Jordan	63
4.4.12	Programme en Matlab de la méthode de Gauss-Jordan	65
4.5	Les méthodes itératives	65
4.5.1	Principe des méthodes itératives	66
4.5.2	Méthode de Jacobi	66
4.5.3	Exercices d'application de la méthode de Jacobi	67
4.5.4	Programme en Matlab de la méthode de Jacobi	68
4.5.5	Méthode de Gauss- Seidel	68
4.5.6	Exercices d'application de la méthode de Gauss- Seidel	69
4.5.7	Programme en Matlab de la méthode de Gauss- Seidel	71
4.6	La convergence des méthodes itératives	71
4.6.1	Rayon Spectral	71
4.7	Série des exercices	72
5	Les méthodes numériques à un pas	76
5.1	Définition les Méthodes à un pas	76
5.2	La méthode d'Euler	76
5.2.1	Exercice d'application de la méthode d'Euler	80
5.2.2	Programme en Matlab de la méthode d'Euler	81
5.3	Méthode d'Euler modifier	81
5.3.1	Exercice d'application de la méthode d'Euler modifier	82
5.3.2	Programme en Matlab de la méthode d'Euler modifier	83
5.4	La méthode de Runge-Kutta d'ordre 2	83
5.4.1	<i>Introduction</i>	83
5.4.2	<i>Définition</i>	83
5.4.3	La formule générale de méthode de Runge-Kutta d'ordre 2	84
5.4.4	Précision de la méthode de Runge-Kutta 2	85
5.4.5	Exercice d'application de la méthode de Runge-Kutta d'ordre 2 . . .	86
5.4.6	Programme en Matlab de la méthode de Runge-Kutta d'ordre 2 . . .	87
5.5	La méthode de Runge-Kutta d'ordre 4	87
5.5.1	Précision de la méthode de Runge-Kutta d'ordre 4	88
5.5.2	Exercice d'application de la méthode de Runge-Kutta d'ordre 4 . . .	88
5.5.3	Programme en Matlab de la méthode de Runge-Kutta d'ordre 4 . . .	89
5.6	Étude générale des méthodes a un pas	90
5.6.1	Consistance :	90
5.6.2	La convergence	92
5.6.3	Stabilité :	92
5.7	Série des exercices	93
	Bibliographie	95

Introduction

Le calcul scientifique est une discipline qui consiste à développer, analyser et appliquer des méthodes relevant de domaines mathématiques aussi variés que l'analyse, l'algèbre linéaire, la géométrie, la théorie de l'approximation, les équations fonctionnelles, l'optimisation ou le calcul différentiel. Le développement de l'analyse numérique, et des mathématiques en général, a été et sera toujours nécessaire pour la résolution des problèmes de plus en

plus complexes posés par la physique, les sciences biologiques, les sciences de l'ingénieur, l'économie et la finance et des autres sciences.

Ce polycopie s'adresse aux étudiants en domaine de science de la matière et sciences de la technologies des filières physique fondamental de deuxième et troisième années et aussi des autres spécialités comme Génie civil, Génie des procédés,..., et rejoint le programme officiel du cours d'analyse numérique de ces filière. Il présente les différentes méthodes avec les fondements théoriques nécessaires, et les exemples d'utilisation de ces méthodes facilitent la conception de leurs diagrammes avec les différents langages.

J'écrirai l'algorithme de chaque méthode pour faciliter de faire le programme avec les différents langages.

Les connaissances nécessaires pour l'assimilation du contenu de cet ouvrage sont les fonctions non linéaires $f(x) = 0$, interpolation, intégration numérique, la résolution du système linéaire et la résolution des équations différentielles.

Toutes les méthodes numériques sont programmées par le biais du "langage" Matlab. Ce dernier est commercialisé par la société MathWorks (<http://www.mathworks.com/>).

Matlab est un langage interprété, son fonctionnement est différent des langages classiques (Fortran, Pascal, ...), dits langages compilés. Un algorithme écrit en langage interprété nécessite pour fonctionner un interprète. Ce dernier est un programme traduisant directement les instructions, en langage machine, au fur et à mesure de leurs exécutions. L'interprète analyse séquentiellement la syntaxe de l'algorithme avant de le dérouler dynamiquement. En revanche, dans un langage compilé, le code source est lu dans un premier temps puis compilé par un compilateur qui le convertit en langage machine directement compréhensible par l'ordinateur. Il en résulte ainsi, qu'un langage interprété sera plus lent qu'un langage compilé à cause de la conversion dynamique de l'algorithme, alors que cette opération est réalisée préalablement pour un langage compilé. Néanmoins, l'un des avantages majeur d'un langage interprété, tient à la facilité de détection d'éventuelles erreurs de programmation.

Chapitre 1

La résolution des équations non linéaires

1.1 Introduction

Dans ce chapitre nous étudions quelques méthodes de résolution d'équations de la forme $f(x) = 0$. Etant donnée une fonction continue $f : [a, b] \rightarrow \mathbb{R}$, nous cherchons des réels x dans $[a, b]$ satisfaisant $f(x) = 0$. On dit aussi, surtout lorsque f est un polynôme, que x est un zéro de f ou encore une racine de f . On connaît des formules permettant de calculer explicitement les racines de tout polynôme de degré < 5 mais de telles formules n'existent pas pour les polynômes de degré supérieur à 5.

L'équation $x - 0.5 \sin(x) - 0.5 = 0$ (cet exemple est emprunté à Dion et Gaudet) admet une racine réelle au voisinage de 0.615.

Les méthodes qui permettent de calculer des valeurs approchées des zéros d'une fonction sont toutes basées sur la notion d'itération.

Elles engendrent une suite de valeurs qui quand tout va bien, converge vers le zéro cherché. Parmi le grand nombre des méthodes disponibles pour chercher une racine d'une fonction f , nous étudierons quatre méthodes :

- 1- Méthode de dichotomie ou bissection.
- 2- Les méthodes de la sécante et de Newton qui consistent à remplacer l'équation $f(x) = 0$ par un polynôme première degré.
- 3- La méthode du point fixe où des approximations successives.

1.2 Préliminaires : séparations des racines

Pour parler sur ce point, on utilise des exemples

Exemple 1.1 résoudre $x - 0.2 \sin(x) - 0.5 = 0$ x réel.

Exemple 1.2 résoudre $\cos(x) = \exp(-x)$ x réel.

Solution 1.1 Le premier travail consiste en une analyse mathématique minimale pour séparer les racines, c'est à dire déterminer des intervalles $[a_i, b_i]$ dans lesquels l'équation considérée a une solution et une seule.

La méthode la plus simple est d'utiliser qu'une fonction continue strictement monotone sur un intervalle $[a_i, b_i]$ et telle que $f(a_i) f(b_i) \leq 0$ a un zéro et un seul dans l'intervalle $[a_i, b_i]$.

Par exemple

$$f_1(x) = x - 0.2 \sin(x) - 0.5$$

$$f_1'(x) = 1 - 0.2 \cos(x) \geq 0$$

pour tout x .

Donc f_1 est strictement croissante sur \mathbb{R} , comme $f_1(0) = -0.5 < 0$, $f_1(\pi) = \pi - 0.5 > 0$, f_1 a un unique zéro dans $[0, \pi]$.

Solution 1.2 La même chose pour deuxième exemple :

$f_2(x) = \cos(x) - \exp(-x)$, la dérivée $f_2'(x) = \sin(x) + \exp(-x)$ est du même type de la fonction, ne peut pas séparer les zéros par cette fonction. Cette situation est bien sur beaucoup plus fréquente, on peut alors :

a- Choisir autre fonction $f_3(x) = \exp(x) \cos(x) - 1$. Les zéros de f_3 sont exactement les solutions de l'exemple 02. D'autre part : $f_3'(x) = \exp(x) (\cos(x) - \sin(x)) = \sqrt{2} \exp(x) (\cos(x - \frac{\pi}{4}))$.

Ainsi f_3 est strictement monotones sur les intervalles du type $[\frac{\pi}{4} + k\frac{\pi}{2}, \frac{\pi}{4} + (k+1)\frac{\pi}{2}]$, k entier. L'étude des signes successifs de $f(\frac{\pi}{4} + k\frac{\pi}{2})$ permet alors de localiser les zéros éventuels.

b- On peut procéder par tâtonnements en s'aidant au maximum d'informations complémentaires disponibles. Ainsi, dans deuxième notre exemple, le tracé des représentations graphiques des fonctions $x \rightarrow \cos(x)$ et $x \rightarrow \exp(-x)$ est particulièrement suggestif.

Dans la suite, nous nous placerons le plus souvent dans le cas où $f : [a, b] \rightarrow \mathbb{R}$ admet un unique zéro dans l'intervalle $[a, b]$.

1.3 Méthode de dichotomie

La méthode de dichotomie, basée sur le théorème de la valeur intermédiaire pour les fonctions continues est, en quelque sorte, ce nous pourrions appeler une méthode de calcul par tâtonnements organisés.

Thorme 1.1 Soit f une fonction continue sur $[a, b]$ et telle que : $f(a) f(b) < 0$, alors f a au moins une racine dans $[a, b]$.

Pour déterminer la solution de l'équation $f(x) = 0$, où f est continue sur $[a, b]$ et $f(a) f(b) < 0$ on procède de la manière suivante :

- 1- Divisons l'intervalle $[a, b]$ en deux intervalles égaux et posons $x_0 = (a + b) / 2$.
- 2- La solution \bar{x} se trouve dans l'un des deux intervalles. Pour savoir lequel, il suffit d'utiliser les conditions suivantes
 - a- Si $f(a) f(x_0) \leq 0$, alors $\bar{x} \in [a, x_0]$
 - b- Si $f(x_0) f(b) \leq 0$, alors $\bar{x} \in [x_0, b]$
- 3- Notons le nouvel intervalle contenant $\bar{x} : [a_1, b_1]$

$$\text{où } a_1 = \begin{cases} a & \text{si } \bar{x} \in [a, x_0] \\ x_0 & \text{si } \bar{x} \in [x_0, b] \end{cases} \quad \text{et } b_1 = \begin{cases} x_0 & \text{si } \bar{x} \in [a, x_0] \\ b & \text{si } \bar{x} \in [x_0, b] \end{cases}$$

en itérant ce procédé, on obtient une suite de valeurs : $x_0 = (a_0 + b_0)/2, x_1 = (a_1 + b_1)/2, \dots, x_n = (a_n + b_n)/2$

La question qui pose maintenant : combien de fois devons nous répéter la division de l'intervalle par deux ?

Après n tours, la dichotomie définit l'intervalle minimum ε_n qui sera le dernier susceptible de modifier effectivement la valeur \bar{x} , c'est à dire : $\varepsilon_n = b_n - a_n = (b_{n-1} - a_{n-1})/2$ par récurrence nous obtenons

$$\varepsilon_n = (b - a) / 2^n \quad (1.1)$$

et si \bar{x} est la racine de l'équation $f(x) = 0$, nous obtenons :

$$|\bar{x} - x_n| \leq b_{n+1} - a_{n+1} = (b - a) / 2^{n+1}$$

Ce qui revient à ce qu'on aille dans les itérations jusqu'à ce que n vérifie l'inégalité

$$n \geq \frac{\log\left(\frac{b-a}{2\varepsilon}\right)}{\log(2)} \quad (1.2)$$

Si $\varepsilon = 10^{-5}, b = 2, a = 1$ alors $n \geq 15.6$

C'est-à-dire le nombre d'itérations nécessaires $n = 16$

1.3.1 Exercice d'application de la méthode de dichotomie

Exercice :

Soit $f(x) = x^3 + 4x^2 - 10, x \in [1, 2]$. Trouver une valeur approchée de la solution \bar{x} de $f(x) = 0$ par la méthode de dichotomie.

Solution :

Soit $f(x) = x^3 + 4x^2 - 10$ est continue sur $[1, 2]$ et $f(1) f(2) < 0$. donc d'après le théorème de la valeur intermédiaire, il existe au moins un racine $\bar{x} \in [1, 2]$. tel que $f(\bar{x}) = 0$. Puisque f est une fonction croissante sur $[1, 2]$, \bar{x} est donc la seule solution de $f(x) = 0$. Cherchons une approximation de celle-ci.

En appliquant l'algorithme de dichotomie, nous obtenons le tableau des valeurs suivant :

n	a_n	b_n	x_n	$f(x_n)$
0	1	2	1.5	2.375
1	1	1.5	1.25	-1.79687
2	1.25	1.5	1.375	0.16211
3	1.25	1.375	1.3125	-0.84839
4	1.125	1.375	1.34375	0.35098
5	1.34375	1.375	1.359375	-0.09641
6	1.359375	1.3671875	1.3671875	0.03236
7	1.359375	1.3671875	1.36328125	-0.03215

Après 8 itérations on peut dire que $x_8 = 1.36328125$ approxime \bar{x} avec une erreur $\varepsilon_7 = 0.00390625$ qui vérifient l'inégalité $|\bar{x} - x_n| \leq (b - a) / 2^{n+1}$.

1- Cette dernière inégalité nous montre que lorsque $n \mapsto +\infty$, $x_n \mapsto \bar{x}$ solution de $f(x) = 0$.

2- Cette même inégalité nous permet d'estimer à l'avance le nombre d'itérations nécessaires pour approcher \bar{x} avec une précision donnée ε .

1.3.2 Programme en Matlab de la méthode de dichotomie

Soit f une fonction continue sur $[a, b]$ telle que $f(a) f(b) < 0$ alors pour trouver une solution approchée de $f(x) = 0$ on procède le programme en Matlab suivant :

```

       clc
        clear all
        n = input('la valeur maximale de nombre d''iteration')
        a = input('la valeur minimale d''intervalle')
        b = input('la valeur maximale d''intervalle')
        epsilon = input('l''erreur de la méthode :')
        syms x
        f = x * exp(x) - 1;
        fa = subs(f, x, a);
        fb = subs(f, x, b);
        for i = 1 : n
            c = (a + b)/2
            fc = subs(f, x, c);
            if fa * fc < = 0
                b = c
            else
                a = c
            end
        if abs(b - a) < = epsilon
            break
        end
        end
        c
        i

```

1.4 Méthode de Newton

On cherche à évaluer numériquement la racine \bar{x} d'une équation $f(x) = 0$, en supposant qu'on dispose d'une valeur grossière x_0 de cette racine.

L'idée est de remplacer la courbe représentative de f par sa tangente au point x_0 :

$$y = f'(x_0)(x - x_0) + f(x_0)$$

L'abscisse x_1 du point d'intersection de cette tangente avec l'axe $y = 0$ est donnée par

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

En générale x_1 est une meilleure approximation de \bar{x} que x_0 , on peut trouver cette expression a partir du théorème de Taylor.

Soit x_n la $n^{\text{ième}}$ itération approximant la solution exacte \bar{x} de $f(x) = 0$, où $f \in C^2[a, b]$ et $f'(x_n) \neq 0$. alors d'après la formule de Taylor on a :

$$f(\bar{x}) = 0 = f(x_n) + (\bar{x} - x_n) f'(x_n) + (\bar{x} - x_n)^2 f''(\xi) / 2$$

où ξ est un nombre compris entre \bar{x} et x_n , cette dernière équation nous donne $\bar{x} = x_n - \frac{f(x_n)}{f'(x_n)}$, c'est à dire que la $(n + 1)^{\text{ième}}$ itération approximant \bar{x} est :

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

Cette suite, si elle converge, doit converger vers la solution \bar{x} de $f(x) = 0$.

1.4.1 Exercice d'application de la méthode de Newton

Exercice :

Soit $f(x) = x^3 - 4x + 1$, $x \in [0, 1]$. Trouver une valeur approchée de la solution \bar{x} de $f(x) = 0$ par la méthode de Newton.

Solution :

Par itération de l'algorithme de Newton

$$x_{n+1} = x_n - \frac{x_n^3 - 4x_n + 1}{3x_n^2 - 4} = \frac{2x_n^3 - 1}{3x_n^2 - 4}$$

i	x_i
0	0
1	0.25
2	0.254098361
3	0.254101688
4	0.254101688

Ceci donne une valeur approchée de \bar{x} à 10^{-9} près environ. Le nombre d'itérations nécessaires pour obtenir une précision de 10^{-9} par la méthode de Newton est typiquement 3 ou 4 (10^{-2^3} , 10^{-2^4}).

Remarque :

- Pratiquement, on arrête le processus itératif des que la différence, en valeur absolue, de deux approximations successives x_i et x_{i-1} ne dépasse pas une certaine précision ε imposée à l'avance. C'est à dire $|\bar{x} - x_n| \leq \varepsilon$.

1.4.2 Programme en Matlab de la méthode de Newton

Soit $f(x) = 0$ et x_0 une approximation de la solution exacte \bar{x} de $f(x) = 0$.

```

clc
clear all
x0 = input('la valeur initiale =')
epsilon = input('l''erreur de la méthode :')
syms x
f = x * exp(x) - 1;
df = diff(f, x);
for i = 1 : 1000000
    f1 = subs(f, x, x0);
    df1 = subs(df, x, x0);
    x1 = x0;
    if(df1 ~ 0)
        x0 = x0 - f1/df1
    if(abs(x1 - x0) < = epsilon
        break
    end
end
end
end

```

1.5 Méthode de la sécante

Dans certaines situations, la dérivée f' est très compliquée ou même impossible à expliciter (c'est le cas par exemple si la fonction f est le résultat d'un algorithme complexe). On ne peut alors utiliser telle quelle la méthode de Newton.

L'idée est de remplacer f' par le taux d'accroissement de f sur un petit intervalle.

Supposons qu'on dispose de deux valeurs approchées x_0, x_1 de la racine \bar{x} de l'équation $f(x) = 0$. Dans un voisinage de l'intervalle $[x_0, x_1]$ on remplace le graphe de $f(x)$ par une droite qui joint les points $(x_0, f(x_0))$ et $(x_1, f(x_1))$ on peut espérer trouver une nouvelle approximation de la racine cherchée en assimilant celle-ci au point x_2 où la droite coupe l'axe des x :

$$x_2 = x_1 - \frac{f(x_1)}{\tau_1} \quad \text{où} \quad \tau_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \quad (1.3)$$

En suite, partant de x_1 et x_2 , on calcule x_3 , etc.

Dans le cas général, nous pouvons écrire la relation suivante :

$$x_{n+1} = x_n - f(x_n) \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}, \quad n = 1, 2, \dots \quad (1.4)$$

où $\tau_n = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$ s'appelle le taux d'accroissement de f sur l'intervalle $[x_n, x_{n-1}]$.

1.5.1 Exercice d'application de la méthode de la sécante

Exercice :

Résoudre l'équation $f(x) = 3x^5 - x^4 - 1 = 0$ par la méthode de la sécante.

Solution :

L'équation $f(x) = 3x^5 - x^4 - 1 = 0$ admet une et une seule racine \bar{x} dans $[0, 1]$. En effet la fonction $f(x) = 3x^5 - x^4 - 1$, nous utilisons l'algorithme de la sécante $x_{n+1} = x_n - f(x_n) \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$, on obtient le tableau des résultats suivant

n	x_{n+1}	$x_n - x_{n-1}$
1	0.15	-1
2	0.5750592	0.4250592
3	0.7787569	0.2036977
4	0.8533380	0.0745812
5	0.8749467	0.0216086
6	0.8824870	0.0003733
7	0.8825820	0.0000950
8	0.8826062	0.0000242

1.5.2 Programme en Matlab de la méthode de la sécante

Soit f une fonction sur l'intervalle $[x_0, x_1]$ alors pour trouver une solution approchée de $f(x) = 0$ on procède :

```

        clc
        clear all
        n = input('le nombre d'interaction maximale')
        x0 = input('la valeur minimale d'intervalle')
        x1 = input('la valeur d'intervalle maximale')
        epsilon = input('erreur d'approximation')
        f = x^2 - x + 1;
        for i = 1 : n
            fx1 = subs(f, x, x1);
            fx0 = subs(f, x, x0);
            x2 = (x0 * fx1 - x1 * fx0) / (fx1 - fx0);
            x0 = x1;
            x1 = x2;
            if x1 - x0 < = epsilon
                break
            end
        end
        i
        x2

```

1.6 Méthode du point fixe (des approximations successives)

Parmi les méthodes de résolution de l'équation $f(x) = 0$ la méthode dite des approximations successives (ou du point fixe) est la plus importe. Son principe est basé sur la construction d'une suite itérative approchant de plus en plus la racine exacte, son premier élément (appelé initialisation) pouvant être n'importe quel point de l'intervalle de travail $[a, b]$.

cette méthode consiste à :

- 1 - Remplacer l'équation $f(x) = 0$ par une équation équivalant $x = g(x)$.
- 2 - Créer la suite de nombres $\{x_n\}_{n=0}^{\infty}$ où $x_n = g(x_{n-1})$ ($n = 1, 2, \dots$) et x_0 est une approximation grossière de \bar{x} solution de $f(x) = 0$.
- 3 - Prendre x_n , lorsque n est suffisamment grand, comme une approximation de la racine \bar{x} solution de $f(x) = 0$.

Thorme 1.2 (du point fixe)

Soit $[a, b]$ un intervalle fermé (non nécessairement borne) et g une fonction de $[a, b]$ dans $[a, b]$. S'il existe un réel $k < 1$ tel que

$$|g(x) - g(y)| \leq k|x - y| \quad x, y \in [a, b]$$

alors l'équation $x = g(x)$ admet une et une seule solution dans $[a, b]$. Cette solution est limite de la suite (x_n) définie par

$$\begin{cases} x_0 \in [a, b] \\ x_{n+1} = g(x_n) \quad n \geq 0 \end{cases}$$

(on est libre de choisir n'importe quel x_0 dans $[a, b]$. De plus, si \bar{x} est la solution de l'équation $x = g(x)$ alors :

$$|\bar{x} - x_n| \leq \frac{k^n}{1-k} |x_1 - x_0| \quad n \geq 1 \quad (1.5)$$

1.6.1 Exercice d'application de la méthode du point fixe

Exercice :

Résoudre $f(x) = x - \sin(x) - \frac{1}{4} = 0$ par la méthode du point fixe.

Solution :

L'équation $f(x) = 0$ équivalant autre équation $x = g(x) = \sin(x) + \frac{1}{4}$, donc l'algorithme s'écrit comme

$$x_{n+1} = g(x_n) = \sin(x_n) + \frac{1}{4},$$

nous choisissons une valeur $x_0 \in [1, 2]$ pour calculer une valeur approchée de la valeur \bar{x} solution de $f(x) = 0$. Les résultats s'écrit dans le tableau suivant

n	x_n
0	1
1	1.091471
2	1.1373063
3	1.1575053
4	1.165804
5	1.1704012
6	1.1709071
7	1.1711041
8	1.1712292

où n est le nombre des itérations et la solution approchée est $x_8 = 1.171292$

Remarques :

1- La suite de la méthode du point fixe n'est pas toujours converge et par conséquent la $n^{\text{ième}}$ itération ne peut pas être une bonne approximation de la racine \bar{x} . Donc pour rendre possible l'application de cette méthode, la fonction g doit satisfaire à certaines conditions que nous allons citer.

2- Soit $g : [a, b] \rightarrow [a, b]$ une fonction dérivable sur $[a, b]$ et telle que : $|g'(x)| \leq k < 1 \forall x \in [a, b]$, alors la suite $\{x_n\}_{n=0}^{\infty}$ définie par $x_{n+1} = g(x_n)$, $n = 1, 2, \dots$, converge indépendamment de la valeur initiale x_0 vers l'unique point fixe \bar{x} de la fonction g .

+ L'inégalité nous permet d'estimer à l'avance le nombre d'itération pour approximer \bar{x} avec une précision donnée ε .

+ Elle nous montre aussi que si $|k|$ est très inférieure à 1 la méthode des approximations successives converge rapidement.

+ Pratiquement, on arrête le processus itératif dès que la différence, en valeur absolue, de deux approximations successives x_n et x_{n-1} ne dépasse pas une certaine précision ε imposée à l'avance, c'est à dire $|\bar{x} - x_n| \leq \varepsilon$.

1.6.2 Programme en Matlab de la méthode du point fixe

Soit l'équation $x = g(x)$ et une approximation grossière x_0 , alors nous pouvons écrire le programme en Matlab suivant :

```

                                clc
                                clear all
                                x0 = input('la valeur initiale =')
                                epsilon = input('l'erreur de la méthode :')
                                syms x
                                g = exp(-x);
                                dg = diff(f, x);
                                for i = 1 : 1000000
                                    g1 = subs(g, x, x0);
                                    dg1 = subs(dg, x, x0);
                                    x1 = x0;
                                if(dg1 ~ = 1)
                                    x0 = g1
                                if(abs(x1 - x0) < = epsilon
                                    break
                                end
                                end
                                end

```

1.7 Série des exercices

Exercice 1- 1 :

Séparer les racines réelles des équations

a- $f(x) = x^4 + 4x + 2$

b- $g(x) = x^3 - 9x^2 + 18x - 1$

c- $h(x) = x - \sin(x)$.

Exercice 1- 2 :

Séparer les racines réelles des équations graphiquement

a- $y(x) = x \log(x) - 1$

b- $k(x) = x \exp(x) - 1$

c- $z(x) = x + 2 \exp(x)$.

Exercice 1- 3 :

Soit l'équation $f(x) = x^4 - 5x - 7$

a)- Trouver la racine négative par la méthode de DICHOTOMIE dans l'intervalle $[-2, -1]$, trouver le nombre d'itération si la précision demandée est $\varepsilon = 0.05$.

b)- Donner trois écritures possibles par la méthode du POINT FIXE en étudiant la convergence pour chaque écriture dans l'intervalle $[-2, -1]$.

Exercice 1- 4 :

01)- Calculer la solution réelle de l'équation $\exp(x) - \frac{1}{x} = 0$ sur l'intervalle $[0.5, 0.7]$ par la méthode de DICHOTOMIE et la méthode de NEWTON, en effectuant quatre itérations (décompositions de l'intervalle).

02)- Comparer les résultats des deux méthodes.

Exercice 1- 5 :

Calculer la solution réelle de l'équation $10^x - 2 \cos(x) = 0$ sur l'intervalle $[0.2, 0.3]$ par la méthode du POINT FIXE, en effectuant quatre itérations, en partant de $x_0 = 0.3$.

Exercice 1- 6 :

Soit la fonction $f(x) = 2x^3 - x - 2$

01)- Vérifier que la fonction f possède une racine réelle $\bar{x} \in [1, 2]$.

02)- Etudier la convergence des trois algorithmes suivants :

a- $x_{n+1} = 2x_n^3 - 2$, b- $x_{n+1} = \frac{2}{2x_n^2 - 1}$, c- $x_{n+1} = \sqrt[3]{1 + \frac{x_n}{2}}$

03)- Si l'une des algorithmes converge, l'utiliser pour déterminer \bar{x} à 10^{-3} près ($x_0 = 1$).

Exercice 1- 7 :

On considère la fonction $f(x) = \exp(x) + 3\sqrt{x} - 2$ sur l'intervalle $[0, 1]$.

1- Montrer qu'il existe un zéro α pour la fonction f dans $[0, 1]$.

2- On veut calculer le zéro α de la fonction f par une méthode de point fixe convernable. En particulier on se donne deux méthodes de point fixe $x = \phi_i, i = 1, 2$, où les fonctions ϕ_1 et ϕ_2 sont définies comme : $\phi_1 = \text{Log}(2 - 3\sqrt{x})$ et $\phi_2 = \frac{(2 - \exp(x))^2}{9}$

Laquelle de ces deux méthodes utiliseriez-vous pour calculer numériquement le zéro α de la fonction f ? Justifiez votre réponse.

3- En utilisant la méthode de la bisection sur l'intervalle $[0, 1]$, pour calculer le zéro α de la fonction f avec une tolérance $\varepsilon = 10^{-10}$.

Exercice 1- 8 :

On se propose de résoudre dans R l'équation : $x \operatorname{ch}(x) = 1$.

Pour cela, on introduit la fonction d'itération g définie sur R par : $g(x) = 1/\operatorname{ch}(x)$ et on définit la suite $(u_n)_n$ par : $u_0 = 0$ et $u_n = g(u_{n-1})$ pour $n \geq 1$.

1- Montrer que $g([0, 1]) \subset [0, 1]$. En déduire que g admet un point fixe dans $[0, 1]$.

2- Calculer g' .

3- i. Montrer que pour tout $u \in R$ on a $\frac{u}{1+u^2} \leq \frac{1}{2}$

3- ii. En déduire que $\forall x \in R |g'(x)| \leq 1/2$

(On rappelle que $\operatorname{ch}^2(x) - \operatorname{sh}^2(x) = 1$).

4- En déduire que $(u_n)_n$ est une suite convergente.

Chapitre 2

Interpolation polynomiale

2.1 Introduction

L'interpolation polynomiale est d'une grande importance dans l'analyse numérique : dérivation et intégration, résolution des équations différentielles,...etc. Jusqu'à maintenant on était concerné par le problème : trouver x tel que $f(x) = 0$, où f est connue : application linéaire ou non linéaire de \mathbb{R}^n vers \mathbb{R}^n . Par contre dans le présent chapitre la fonction f n'est pas connue, sauf en certains points, sur tout le domaine de définition de f . notre but est donc construire une "bonne approximation" de f à partir d'un nombre fini d'informations sur celle-ci voici un problème d'interpolation : un certain pays recense sa population tous les dix ans. La table ci-dessous résume les résultats des recensements survenus dans la période :

Année	1960	1970	1980	1990	2000	2010
Populations en millions	12.567	14.658	17.354	20.132	21.456	24.089

Question : En analysant ces données, peut-on savoir quelle était la population en 2020 ou quelle sera la population en l'an 2030 ? Ce problème est un problème d'interpolation.

Les fonctions les plus faciles à évaluer numériquement sont les fonctions polynômes. Il est important de savoir approximer une fonction arbitraire par des polynômes. Dans ce cadre, l'un des outils de base est la méthode d'interpolation de Lagrange.

2.2 Méthode d'interpolation de Lagrange

Soit f une fonction continue sur l'intervalle $[a, b]$. On se donne $(n + 1)$ points $x_0, x_1, x_2, \dots, x_n$ dans $[a, b]$, deux à deux distincts, non nécessairement rangés par ordre croissant. on a donc $f(x_i) = y_i, i = 0, 1, 2, \dots, n$

2.2.1 Existence et unicité du polynôme d'interpolation

Problème :

Existe-t-il un polynôme de degré inférieur ou égale à n tel que

$$P_n(x_i) = f(x_i) \quad \forall i = 0, 1, \dots, n \quad (2.1)$$

un tel polynôme sera appelé polynôme d'interpolation (de Lagrange) de f aux points x_0, x_1, \dots, x_n .

Posons

$$L_i(x) = \prod_{j \neq i}^n \frac{(x - x_j)}{(x_i - x_j)} \quad 0 \leq i \leq n \quad (2.2)$$

où le produit est effectué sur les indices j tels que $0 \leq j \leq n, j \neq i$. Il est clair que

$$L_i(x_j) = \delta_{ij} = \begin{cases} 1 & \text{si } j = i \\ 0 & \text{si } j \neq i \end{cases} \quad (2.3)$$

Le problème ci dessus admet donc au moins une solution

$$P_n(x) = \sum_{i=0}^n f(x_i) L_i(x) \quad (2.4)$$

Thorme 2.1 *Le problème d'interpolation $P_n(x_i) = f(x_i) \quad \forall i = 0, 1, \dots, n$, admet une solution et une seule, donnée par la formule précédente (2.4). Il reste à prouver l'unicité. Supposons que $Q_n(x)$ soit une autre solution du problème. Alors $P_n(x_i) = Q_n(x_i) = f(x_i)$, donc x_i racine de $Q_n - P_n$. Par suite le problème $w(x) = \prod_{j=0}^n (x - x_j)$ divise $Q_n - P_n$. Comme $\deg(w(x)) = n + 1$ et $\deg(Q_n - P_n) = n$, la seule possibilité est que $Q_n - P_n = 0$ donc $Q_n = P_n$.*

Remarques :

1 - On a $w(x) = (x - x_i) \prod_{i=0, j \neq i}^n (x - x_j)$ d'où $w'(x) = \prod_{j \neq i}^n (x - x_j)$, alors les polynômes de Lagrange s'écrit comme :

$$L_i(x) = \frac{w(x)}{(x - x_i) w'(x_i)} \quad 0 \leq i \leq n \quad (2.5)$$

2 - Pour démontrer le théorème précédent, on peut également poser $P_n(x) = \sum_{i=0}^n a_i x^i$ et résoudre un système linéaire de $(n + 1)$ équations

$$\sum_{j=0}^n a_j x_i^j = f(x_i) \quad 0 \leq i \leq n \quad (2.6)$$

en les $(n + 1)$ inconnues a_0, a_1, \dots, a_n . L déterminant du système est un déterminant dit de Van der Monde

$$\Delta = \begin{vmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{vmatrix}$$

Il s'agit de montrer que $\Delta \neq 0$ car les x_i sont distincts. D'où, le polynôme d'interpolation existe et unique.

2.2.2 Détermination pratique des polynômes de Lagrange

La détermination de $(n + 1)$ polynômes $L_i(x)$ peut être conduite de la façon suivante : on dresse le tableau carré

$$\begin{array}{cccccc} (x - x_0) & (x_0 - x_1) & (x_0 - x_2) & (x_0 - x_3) & \cdots & (x_0 - x_n) \\ (x_1 - x_0) & (x - x_1) & (x_1 - x_2) & (x_1 - x_3) & \cdots & (x_1 - x_n) \\ (x_2 - x_0) & (x_2 - x_1) & (x - x_2) & (x_2 - x_3) & \cdots & (x_2 - x_n) \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ (x_n - x_0) & (x_n - x_1) & (x_n - x_2) & (x_n - x_3) & \cdots & (x - x_n) \end{array}$$

et on voit que

$$L_i(x) = \frac{\text{Le produit des termes diagonaux du tableau}}{\text{Le produit des termes de la } (i + 1)\text{ième ligne du tableau}}$$

2.2.3 Exercices d'application de la méthode de l'interpolation de Lagrange

Exercice 1 :

- Calculer a) $\det \begin{bmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{bmatrix}$, b) $\det \begin{bmatrix} 1 & 2 & 4 \\ 1 & 1 & 1 \\ 1 & 3 & 9 \end{bmatrix}$.

Solution 1 :

$$\text{a)- } \det \begin{bmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{bmatrix} = \prod_{i>j} (x_i - x_j) = (x_1 - x_0)(x_2 - x_0)(x_2 - x_1)$$

$$\text{b)- } \det \begin{bmatrix} 1 & 2 & 4 \\ 1 & 1 & 1 \\ 1 & 3 & 9 \end{bmatrix} = (1 - 2)(3 - 1)(3 - 2) = -2.$$

Exercice 2 :

Déterminer les 5 polynômes $L_i(x)$ attachés aux valeurs $x_0 = 0$, $x_1 = 4$, $x_2 = -4$, $x_3 = 2$, et $x_4 = 1$.

Solution 2 :

Considérons le tableau ci-après

$$\begin{array}{ccccc} x & -4 & 4 & -2 & -1 \\ 4 & x - 4 & 8 & 2 & 3 \\ -4 & -8 & x + 4 & -6 & -5 \\ 2 & -2 & 6 & x - 2 & 1 \\ 1 & -3 & 5 & -1 & x - 1 \end{array}$$

Nous avons alors

$$\begin{aligned}
L_0(x) &= \frac{x(x-4)(x+4)(x-2)(x-1)}{x(4)(-4)(2)(1)} = -\frac{1}{32}x^4 + \frac{3}{32}x^3 + \frac{7}{16}x^2 - \frac{3}{2}x - 1. \\
L_1(x) &= \frac{x(x-4)(x+4)(x-2)(x-1)}{(4)(x-4)(8)(2)(3)} = \frac{1}{192}x^4 + \frac{1}{192}x^3 - \frac{5}{96}x^2 + \frac{1}{24}x. \\
L_2(x) &= \frac{x(x-4)(x+4)(x-2)(x-1)}{(-4)(-8)(x+4)(-6)(-5)} = \frac{1}{960}x^4 - \frac{7}{960}x^3 + \frac{7}{480}x^2 - \frac{1}{120}x. \\
L_3(x) &= \frac{x(x-4)(x+4)(x-2)(x-1)}{(2)(-2)(6)(x-2)(1)} = -\frac{1}{24}x^4 + \frac{1}{24}x^3 + \frac{2}{3}x^2 - \frac{2}{3}x. \\
L_4(x) &= \frac{x(x-4)(x+4)(x-2)(x-1)}{(1)(-3)(5)(-1)(x-1)} = \frac{1}{15}x^4 - \frac{2}{15}x^3 + \frac{16}{15}x^2 - \frac{32}{15}x.
\end{aligned}$$

Exercice 3 : Cas particuliers des polynômes d'interpolation pour $n = 1$ et $n = 2$.

Solution 3 :

Pour $n = 1$

Le polynôme d'interpolation d'ordre 1 s'écrit

$$P_1(x) = y_0 \frac{x - x_1}{x_0 - x_1} + y_1 \frac{x - x_0}{x_1 - x_0}$$

c'est l'équation d'une droite passant par les points (x_0, y_0) et (x_1, y_1) donnés.

Pour $n = 2$

$$P_2(x) = y_0 \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + y_1 \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} + y_2 \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}$$

qui l'équation d'une parabole qui passe par les trois points (x_0, y_0) , (x_1, y_1) et (x_2, y_2) .

Exercice 4 :

Construire le polynôme d'interpolation de Lagrange de la fonction $y = f(x) = \sin(\pi x)$ aux points $x_0 = 0$, $x_1 = 1/6$ et $x_2 = 1/2$.

Solution 4 :

Dans ce cas $n = 2$ et nous devons chercher un polynôme de degré inférieur ou égal à 2 de la forme :

$$P_2(x) = y_0 \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + y_1 \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} + y_2 \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}$$

comme $y_0 = f(x_0) = 0$, $y_1 = f(x_1) = 1/2$ et $y_2 = f(x_2) = 1$; alors $P_2(x) = \frac{7}{2}x - 3x^2$.

2.2.4 Programme en Matlab de la méthode de l'interpolation de Lagrange

Soit $(n + 1)$ points $(x_0, f_0), (x_1, f_1), \dots,$ et (x_n, f_n) données pour déterminer le polynôme d'interpolation de degré inférieur ou égal à n .

```

    clc
    clear all
    x = input('le vecteur des abscices')
    f = input('le vecteur des images pour chaque abscice')
    syms t
    n = length(x);
    s = 0;
    for i = 1 : n
        p = 1;
        for j = 1 : n
            if j ~ = i
                p = p * (t - x(j))/(x(i) - x(j));
            end
        end
        s = s + p * f(i);
    end
    pn(t) = s

```

2.3 Méthode d'interpolation de Newton

Nous pouvons utiliser autre méthode plus simple et efficace permettant de calculer les polynômes d'interpolation de f .

Soit P_n le polynôme d'interpolation de f aux $x_0, x_1, x_2, \dots, x_n$.

2.3.1 Différence divisée

On désigne par $\delta [x_0, x_1, x_2, \dots, x_n] f$ le coefficient directeur du polynôme P_n (=coefficient de x^n dans $P_n(x)$).

alors $P_n - P_{n-1}$ est un polynôme de degré $\leq n$ s'annulant aux points $x_0, x_1, x_2, \dots, x_{n-1}$ et admettant $\delta [x_0, x_1, x_2, \dots, x_n] f$ pour coefficient directeur. Par suite :

$$P_n(x) - P_{n-1}(x) = \delta [x_0, x_1, x_2, \dots, x_n] f (x - x_0)(x - x_1) \cdots (x - x_{n-1})$$

Comme $P_0(x) = f(x_0)$, on en déduit la formule fondamentale

$$P_n(x) = f_0(x) + \sum_{k=1}^n \delta [x_0, x_1, x_2, \dots, x_k] f (x - x_0)(x - x_1) \cdots (x - x_{k-1}) \quad (2.7)$$

Pour pouvoir comment exploiter cette formule, il reste bien entendu à évaluer les coefficients $\delta [x_0, x_1, x_2, \dots, x_k] f$, on utilise une relation de récurrence sur le nombre k de point x_i , où $\delta [x_0] f = f(x_0)$

pour $k \geq 1$, on a :

$$\delta [x_0, x_1, x_2, \dots, x_k] f = \frac{\delta [x_1, x_2, \dots, x_k] f - \delta [x_0, x_1, x_2, \dots, x_{k-1}] f}{x_k - x_0} \quad (2.8)$$

A cause de cette formule, la quantité $\delta [x_0, x_1, x_2, \dots, x_k] f$ est appelée différence divisée d'ordre k de f aux points $x_0, x_1, x_2, \dots, x_k$.

Pour calculer la différence divisée d'ordre n de la fonction f aux points $x_0, x_1, x_2, \dots, x_n$, on forme le tableau suivant en appliquant la formule de récurrence précédente colonne après colonne

x_i	f_i	δ_1	δ_2	δ_3	\dots	\dots	δ_n
x_0	f_0	$\delta [x_0, x_1] f$	$\delta [x_0, x_1, x_2] f$	$\delta [x_0, x_1, x_2, x_3] f$	\dots	\dots	$\delta [x_0, x_1, x_2, \dots, x_n] f$
x_1	f_1	$\delta [x_1, x_2] f$	$\delta [x_1, x_2, x_3] f$	$\delta [x_1, x_2, x_3, x_4] f$	\dots		
x_2	f_2	$\delta [x_2, x_3] f$	$\delta [x_2, x_3, x_4] f$	$\delta [x_2, x_3, x_4, x_5] f$			
x_3	f_3	$\delta [x_3, x_4] f$	$\delta [x_3, x_4, x_5] f$				
\vdots	\vdots	\vdots					
\vdots	\vdots	$\delta [x_{n-1}, x_n] f$					
x_n	f_n						

D'après la formule (2.7), nous pouvons écrire la formule suivante pour $(n + 1)$ points $x_0, x_1, x_2, \dots, x_n$

$$P_n(x) = f_0(x) + \delta [x_0, x_1] f (x - x_0) + \delta [x_0, x_1, x_2] f (x - x_0)(x - x_1) + \dots + \delta [x_0, x_1, x_2, \dots, x_n] f (x - x_0)(x - x_1) \dots (x - x_{n-1}) \quad (2.9)$$

cette formule s'appelle la formule de Newton pour le polynôme d'interpolation.

2.3.2 Cas où les points d'interpolation sont équidistants

On considère la subdivision de l'intervalle $[a, b]$ de pas constant $h = \frac{b-a}{n}$. Les points d'interpolation sont donc $x_i = a + i h$, $0 \leq i \leq n$.

Différences finies progressives

On note $f_i = f(x_i)$ les valeurs de f correspondante et on introduit un opérateur noté Δ , appelée opérateur aux différences finies progressives défini par :

$$\begin{aligned} \Delta f_i &= f_{i+1} - f_i \\ \Delta^2 f_i &= \Delta(\Delta f_i) & 0 \leq i \leq n-1 \\ &= \Delta f_{i+1} - \Delta f_i \end{aligned}$$

Dans le cas général nous pouvons écrire :

$$\Delta^k f_i = \Delta^{k-1} f_{i+1} - \Delta^{k-1} f_i \quad k \geq 1 \quad \text{et} \quad 0 \leq i \leq n - k$$

où l'on convient que $\Delta^0 f_i = f_i$, $0 \leq i \leq n$.

Il est alors facile de montrer par récurrence que les différences divisées sont données par :

$$\delta [x_i, x_{i+1}, x_{i+2}, \dots, x_{i+k}] f = \frac{\Delta^k f_i}{k! h^k}$$

Récrivons avec ces notations la formule fondamentale (2.7). Pour $x \in [a, b]$, effectuons le changement de variable $x = a + s h$, $s \in [0, n]$.

On a alors

$$(x - x_0)(x - x_1) \cdots (x - x_{k-1}) = h^k s(s-1)(s-2) \cdots (s-(k-1))$$

On obtient la formule de Newton suivante :

dans laquelle $s = (x - a) / h$

$$\begin{aligned} P_n(x) &= \sum_{k=0}^n \frac{\Delta^k f_0}{k!} s(s-1)(s-2) \cdots (s-(k-1)) \\ &= f_0 + s \Delta f_0 + s(s-1) \frac{\Delta^2 f_0}{2!} + s(s-1)(s-2) \frac{\Delta^3 f_0}{3!} + \cdots + s(s-1)(s-2) \cdots (s-(n-1)) \end{aligned}$$

Pour calculer les différences finis progressives $\Delta^k f_0$, nous utilisons le tableau suivant :

x_i	f_i	Δ_1	Δ_2	Δ_3	\cdots	\cdots	Δ_n
x_0	f_0	$\Delta^2 f_0$	$\Delta^2 f_0$	$\Delta^3 f_0$	\cdots	\cdots	$\Delta^n f_0$
x_1	f_1	Δf_1	$\Delta^2 f_1$	$\Delta^3 f_1$	\cdots		
x_2	f_2	Δf_2	$\Delta^2 f_2$	\vdots			
x_3	f_3	Δf_3	\vdots	$\Delta^3 f_{n-3}$			
\vdots	\vdots	\vdots	$\Delta^2 f_{n-2}$				
\vdots	\vdots	Δf_{n-1}					
x_n	f_n						

Différences finies régressives

Nous pouvons écrire aussi ce polynôme par la relation des différences finies régressives.

On note l'opérateur $\nabla f_i = f_i - f_{i-1}$ $1 \leq i \leq n$

dans le cas général nous pouvons écrire :

$$\begin{aligned} \nabla^k f_i &= \nabla^{k-1} (\nabla f_i) \\ &= \nabla^{k-1} f_i - \nabla^{k-1} f_{i-1} \quad 1 \leq i \leq n \end{aligned}$$

dans le cas général nous pouvons écrire :

$$\Delta^k f_i = \Delta^{k-1} f_{i+1} - \Delta^{k-1} f_i \quad k \geq 1 \quad \text{et} \quad k \leq i \leq n$$

où l'on convient que $\nabla^0 f_i = f_i$, $0 \leq i \leq n$.

Il est facile de montrer par récurrence que les différences divisées sont données par :

$$\delta [x_i, x_{i+1}, x_{i+2}, \dots, x_{i+k}] f = \frac{\nabla^k f_{i+k}}{k! h^k}$$

Récrivons avec ces notations la formule fondamentale (2.7). Pour $x \in [a, b]$, effectuons aussi le même changement de variable précédent $x = a + s h$, $s \in [0, n]$.

On a alors

$$\begin{aligned}
 P_n(x) &= \sum_{k=0}^n \frac{\nabla^k f_n}{k!} (s-n)(s-n+1)\cdots(s-n+k) \\
 &= f_0 + (s-n)\nabla f_n + (s-n)(s-n+1)\frac{\nabla^2 f_n}{2!} + \cdots + (s-n)(s-n+1)\cdots(s-n)\frac{\nabla^n f_n}{n!}
 \end{aligned}$$

2.3.3 Exercice d'application de la méthode de l'interpolation de Newton

Exercice :

Soit $(n+1)$ points $(x_0, f_0), (x_1, f_1), \dots,$ et (x_n, f_n) données dans le tableau suivant :

x_i	-1	2	4	5
f_i	-2	43	213	376

déterminer le polynôme d'interpolation de degré inférieure ou égal à n par la méthode de Newton.

solution :

La table de différences divisées s'écrit :

x_i	f_i	δ_1	δ_2	δ_3
-1	-2	15	14	2
2	43	85	26	
4	213	163		
5	376			

On en déduit le polynôme d'interpolation de degré inférieure ou égal à n par la méthode de Newton :

$$\begin{aligned}
 P_3(x) &= -2 + 15(x+1) + 14(x+1)(x-2) + 2(x+1)(x-2)(x-4) \\
 &= 2x^3 + 4x^2 + 5x + 1
 \end{aligned}$$

2.3.4 Programme en Matlab de la méthode de l'interpolation de Newton

Soit $(n+1)$ points $(x_0, f_0), (x_1, f_1), \dots,$ et (x_n, f_n) données pour déterminer le polynôme d'interpolation de degré inférieure ou égal à n .

```

        clc
        clear all
        close all
        x = input('le vecteur des abscices')
        f = input('le vecteur des images pour chaque abscice')
        syms t
        n = length(x);
        for i = 1 : n
        y(i,1) = f(i)
        end
        for j = 2 : n
        for i = 1 : n - j + 1
        y(i,j) = (y(i+1,j-1) - y(i,j-1))/(x(i+j-1) - x(i));
        end
        end
        s = y(1,1)
        for j = 2 : n
        p = 1;
        for k = 1 : j - 1
        p = p * (t - x(k));
        end
        s = s + p * y(1,j);
        end
        pn(t) = s

```

2.4 Formule d'erreur d'interpolation

On va estimer l'erreur mathématique noté $E(x)$

$$E(x) = |f(x) - P_n(x)| \quad (2.10)$$

dans le cas où la fonction est supposée $(n+1)$ fois dérivable sur l'intervalle de travail $[a, b]$ et où sa dérivée d'ordre $(n+1)$ est bornée.

Les points $x_0, x_1, x_2, \dots, x_n$ sont les racines de la fonction $E(x)$ alors la formule de l'erreur s'écrit comme :

$$E(x) = C(x - x_0)(x - x_1) \cdots (x - x_n)$$

nous calculons la constante C d'après la dérivée d'ordre $(n+1)$ de la formule (2.10)

$$E^{(n+1)}(x) = f^{(n+1)}(x) - P_n^{(n+1)}(x)$$

cette formule est équivalent de $C(n+1)! = f^{(n+1)}(\xi)$

et donc, par application du théorème de Rolle à la fonction E dans les intervalles $[x_0, x_1]$, $[x_1, x_2]$, ..., $[x_{n-1}, x_n]$, il existe des points $(\alpha'_i)_{i=0, \dots, n-1}$, telle que $E'(\alpha'_i) = 0$.

On réitère le théorème de Rolle pour la fonction E' , puisque $E'(\alpha'_0) = E'(\alpha'_1) = E'(\alpha'_2) = \dots = E'(\alpha'_{n-1}) = 0$, pour avoir cette fois n points $(\alpha''_i)_{i=0, \dots, n-2}$ tels que $E''(\alpha''_0) = E''(\alpha''_1) = E''(\alpha''_2) = \dots = E''(\alpha''_{n-2}) = 0$, puis, de proche en proche, il existe un point $\xi \in]a, b[$ telle que

$$E^{(n+1)}(\xi) = C(n+1)! = f^{(n+1)}(\xi)$$

alors la constante $C = \frac{f^{(n+1)}(\xi)}{(n+1)!}$, A partir de l'expression de la constante C nous écrivons l'expression de l'erreur

$$E(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x-x_0)(x-x_1)\cdots(x-x_n) \quad \text{où } \xi \in]a, b[\quad (2.11)$$

Cette expression est l'erreur de l'interpolation du polynôme.

2.5 Série des exercices

Exercice 2 - 1 :

Donner le polynôme d'interpolation de Lagrange de la fonction aux points indiqués :

$$\begin{array}{ll} f(x) = \exp(x), & x_0 = -1, x_1 = 1 \text{ et } x_2 = 1 \\ f(x) = \sin(\pi x), & x_0 = 0, x_1 = 1/2 \text{ et } x_2 = 1 \\ f(x) = x^3, & x_0 = -1, x_1 = 0 \text{ et } x_2 = 1 \\ f(x) = 3x + 2, & x_0 = 0, x_1 = 1/2 \text{ et } x_2 = 1 \end{array}$$

Exercice 2 - 2 :

On considère la fonction définie par :

x_i	0	1	2	3
f_i	16	9	0	-5

- Déterminer le polynôme d'interpolation de f par la méthode de Lagrange.
- Construire les tables des différences divisées et des différences finies de f .
- En déduire le polynôme d'interpolation par la méthode de Newton.

Exercice 2 - 3 :

Montrer que les polynômes de Lagrange vérifient les propriétés suivantes :

$$\begin{array}{l} 1 - \quad L_i(x_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \\ 2 - \quad \sum_{i=1}^n L_i(x) = 1 \\ 3 - \quad L_i(x) = \frac{w(x)}{(x-x_i)w'(x)} \quad \text{où } w(x) = \prod_{i=0}^n (x-x_i) \end{array}$$

Exercice 2 - 4 :

Soit une fonction dont on connaît les valeurs f_0, f_1, \dots, f_n en $(n + 1)$ points x_0, x_1, \dots, x_n .
Etablir par récurrence les expressions suivantes :

$$\frac{\nabla^k f_j}{k!h^k} = \delta(x_{j-k}, x_{j-k+1}, \dots, x_j) f$$

$$\delta(x_0, x_1, \dots, x_n) f = \sum_{k=0}^n \frac{f(x_k)}{\prod_{j=0 \text{ et } j \neq k}^n (x_k - x_j)}$$

Exercice 2 - 5 :

On considère la fonction $f(x) = \exp(-x/10)$ définie sur l'intervalle $[0, 3]$ par la table de valeurs :

x_i	0	1	2	3
f_i	1	0.904837	0.818730	0.740818

1. Calculer à l'aide de la méthode d'interpolation de Lagrange, la valeur de $f(1.5)$.
2. Calculer l'erreur d'interpolation que l'on commet alors.
3. Déterminer le polynôme d'interpolation de degré 3 passant par les points donnés ci-dessus par la formule de Newton progressive.

Exercice 2 - 6 :

1)- Donner le polynôme d'interpolation de Lagrange en points $-1, 0, 1, 2$ de la fonction $f(x) = \cos(\pi x - \pi)$.

2)- Montrer que les polynômes de Lagrange satisfont aux $(n + 1)$ équations suivantes :

$$\sum_{i=0}^n x_i^k L_i(x) = x^k, \quad k = 0, \dots, n \quad (2.12)$$

où les $x_i, i = 0, \dots, n$, sont $(n + 1)$ valeurs données.

3)- En notant par $\pi(x) = \prod_{j=0}^n (x - x_j)$, montrer que les coefficients de Lagrange sont donnés par : $L_j(x) = \frac{\pi(x)}{(x - x_j) \pi'(x_j)}$.

Chapitre 3

Méthodes des intégrations numériques

3.1 Introduction

Dans les méthodes d'intégration, l'intégrale d'une fonction continue sur un intervalle borné $[a, b]$ est remplacée par une somme finie. Le choix de la subdivision de l'intervalle d'intégration et celui des coefficients qui interviennent dans la somme approchant l'intégrale sont des critères essentiels pour minimiser l'erreur. Ces méthodes se répartissent en deux grandes catégories : les méthodes des composées dans lesquelles la fonction f est remplacée par un polynôme d'interpolation sur chaque intervalle élémentaire $[x_i, x_{i+1}]$ de la subdivision et les méthodes de Gauss fondée sur les polynômes orthogonaux pour lesquelles les points de la subdivision sont imposés.

3.2 Méthode des trapèzes

3.2.1 Principe de la méthode

On remplace f sur chaque segment de la subdivision par la fonction affine du type $Ax + B$ qui prend les mêmes valeurs que f aux deux extrémités de ce segment.

La méthode d'approximation d'intégrale ainsi dénommée repose sur le calcul de l'aire d'un trapèze sur l'intervalle $[a, b]$

$$\int_a^b f(x)dx \simeq \frac{1}{2}(f(a) + f(b))(b - a) \quad (3.1)$$

On utilise l'interpolation de Lagrange entre deux points $(a, f(a)), (b, f(b))$

$$P_1(x) = f(a) L_0(x) + f(b) L_1(x) \quad (3.2)$$

on intègre les deux membres de l'équation précédente

$$\int_a^b f(x)dx = f(a) \int_a^b L_0(x)dx + f(b) \int_a^b L_1(x)dx + E_T \quad (3.3)$$

calculons maintenant les deux intégrales. On a par définition des polynômes d'interpolation de Lagrange

$$\{L_0(x) = \frac{x-b}{a-b}L_1(x) = \frac{x-a}{b-a}$$

l'intégration les deux polynômes $L_0(x)$ et $L_1(x)$ est donné par

$$\int_a^b L_0(x)dx = \frac{1}{a-b} \int_a^b (x-b)dx = \frac{b-a}{2},$$

$$\int_a^b L_1(x)dx = \frac{1}{b-a} \int_a^b (x-a)dx = \frac{b-a}{2},$$

donc l'expression 3.3 s'écrit

$$\int_a^b f(x)dx = \frac{b-a}{2}(f(a) + f(b)) + E_T, \quad (3.4)$$

où E_T est l'erreur de la méthode, cette formule s'appelle formule des trapèzes.

3.2.2 Formule généralisée des trapèzes

Considérons alors une subdivision de l'intervalle $[a, b]$ de pas h . cette subdivision est définie par n telle que $h = \frac{b-a}{n}$ et par la suite de réels : " $a, a+h, \dots, a+kh, \dots, a+nh = b$ " sur chaque intervalle $[a+kh, a+(k+1)h]$ où k est un entier compris entre 0 et $(n-1)$

$$\int_a^b f(x)dx = \int_{x_0}^{x_1} f(x)dx + \int_{x_1}^{x_2} f(x)dx + \dots + \int_{x_{n-1}}^{x_n} f(x)dx = \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} f(x)dx$$

d'après l'utilisation de la formule des trapèzes entre les deux points (x_i, x_{i+1})

$$\int_{x_i}^{x_{i+1}} f(x)dx = \frac{h}{2}(f(x_i) + f(x_{i+1})) + E_{T_i},$$

nous pouvons écrire :

$$\begin{aligned} \int_a^b f(x)dx &= \frac{h}{2}(f(x_0) + f(x_1)) + \frac{h}{2}(f(x_1) + f(x_2)) + \dots + \frac{h}{2}(f(x_{n-1}) + f(x_n)) + nE_{T_i} \\ &= \frac{h}{2}(f(x_0) + 2 \sum_{i=1}^{n-1} f(x_i) + f(x_n)) + E_T, \end{aligned} \quad (3.5)$$

cette formule s'appelle formule généralisée des trapèzes.

3.2.3 L'erreur de la méthode des trapèzes

La fonction f qui s'écrit :

$$f(x) = P_1(x) + \varepsilon_1(x)$$

On a

$$\int_a^b f(x)dx = f(x_0) \int_a^b L_0(x)dx + f(x_1) \int_a^b L_1(x)dx + \int_a^b \varepsilon_1(x)dx$$

tel que $\varepsilon_n(x) = \frac{f^{(n+1)}(\alpha)}{(n+1)!} w(x)$ / $\alpha \in [a, b]$ (voir chapitre d'interpolation)

avec $w(x) = \prod_{i=0}^n (x - x_i)$

Dans ce cas la méthode des trapèzes ($n = 1$) alors :

$$\varepsilon_1(x) = \frac{f^{(2)}(\alpha)}{2!} (x - x_0)(x - x_1)$$

$$E_T = \int_{x_0}^{x_1} \varepsilon_1(x)dx + \int_{x_1}^{x_2} \varepsilon_2(x)dx + \cdots + \int_{x_{n-1}}^{x_n} \varepsilon_n(x)dx \simeq n \int_{x_0}^{x_1} \varepsilon_1(x)dx$$

$$E_T = \frac{f^{(2)}(\alpha)}{2} \cdot n \int_{x_0}^{x_1} (x - x_0)(x - x_1)dx$$

alors

$$E_T = -\frac{f^{(2)}(\alpha)}{12n^2} (b - a)^3 \quad (3.6)$$

Dans le cas générale, on ne peut pas calculer E_T puisque le variable α n'est pas définit, à partir de ça nous utilisons le majorant de l'erreur

si f est une fonction de classe C^2 sur $[a, b]$

$$\left| \int_a^b f(x)dx - \frac{b-a}{2n} \sum_{i=0}^{n-1} (f(x_i) + f(x_{i+1})) \right| \leq \frac{(b-a)^3}{12n^2} M_2$$

où $M_2 = \sup_{[a,b]} |f^{(2)}(\alpha)|$

alors

$$\int_a^b f(x)dx = \frac{h}{2} (f(x_0) + 2 \sum_{i=1}^{n-1} f(x_i) + f(x_n)) + \frac{(b-a)^3}{12n^2} M_2 \quad (3.7)$$

3.2.4 Exercice d'application de la méthode des trapèzes

Exercice :

On considère l'intégrale

$$I = \int_1^2 \frac{1}{x} dx$$

1. Calculer la valeur exacte de I .
2. Évaluer numériquement cette intégrale par la méthode des trapèzes avec $n = 3$ sous-intervalles.

Solution :

1. La valeur exacte est $I = [\ln(x)]_{x=1}^{x=2} = \ln(2) - \ln(1) = \ln(2)$.
2. La méthode des trapèzes composite à $(n + 1)$ points pour calculer l'intégrale d'une fonction f sur l'intervalle $[a, b]$ s'écrit

$$\int_a^b f(t) dt = \frac{h}{2} (f(a) + 2 \sum_{i=1}^{n-1} f(a + ih) + f(b)) \quad \text{avec} \quad h = \frac{b-a}{n}.$$

Ici on a $f(x) = \frac{1}{x}$, $a = 1$, $b = 2$, $n = 3$ d'où $h = \frac{1}{3}$ et on obtient

$$I \simeq \frac{1}{6} (f(1) + 2f(1 + 1/3) + 2f(1 + 2/3) + f(2)) = \frac{1}{6} (1/1 + 2 * 3/4 + 2 * 3/5 + 1/2) = \frac{21}{30} = 0.7. \quad \blacksquare$$

3.2.5 Programme en Matlab de la méthode des trapèzes

```

clc
clear all
close all

f = input('les valeurs de la fonction pour chaque abscice')
a = input('la valeur minimale d'intervalle')
b = input('la valeur maximale d'intervalle')
n = length(f) - 1
for i = 0 : n
    x(i + 1) = a + i * h
end

h = (b - a)/n
s = 0;
for i = 2 : n - 1
    s = s + f(i)
end

Itrap = (h/2) * (f(1) + 2 * s + f(n))

```

3.3 Méthode de Simpson

3.3.1 Principe de la méthode

On remplace f sur chaque segment $[x_i, x_{i+1}]$ par un polynôme P de degré 2 au plus, qui prend les mêmes valeurs que f aux deux extrémités et au milieu de ce segment.

La méthode de Simpson consiste à grouper trois points consécutifs de la courbe M_i, M_{i+1}, M_{i+2} et de remplacer l'arc de la courbe passant par ces trois points par un arc de parabole.

Dans cette méthode on utilise le polynôme d'interpolation de Lagrange de degré 2, ce dernier est donné par :

$$P_2(x) = \sum_{i=0}^2 L_i(x) f(x_i) \quad (3.8)$$

où $L_i(x)$: sont les polynômes de Lagrange qui défini par :

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{(x - x_j)}{(x_i - x_j)} \quad (3.9)$$

Pour calculer l'intégrale de f en passant de calculer l'intégrale de polynôme d'interpolation de Lagrange de degré 2 :

$$\int_a^b f(x) dx = \int_a^b P_2(x) dx + \int_a^b \varepsilon_2(x) dx$$

on pose : $a = x_0, b = x_2,$

$$\begin{cases} x_1 = a + h \\ x_2 = a + 2h \end{cases}$$

alors $x_1 = \frac{x_0 + x_2}{2} = \frac{a + b}{2}$

on a :

$$f(x) = P_2(x) + \varepsilon_2(x)$$

$$P_2(x) = \sum_{i=0}^2 L_i(x) f(x_i)$$

On va calculer les polynômes de Lagrange $L_0(x), L_1(x)$ et $L_2(x)$

$$L_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} = \frac{\left(x - \frac{a+b}{2}\right)(x - b)}{2h^2}$$

$$L_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} = \frac{(x - a)(x - b)}{-h^2}$$

$$L_2(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} = \frac{(x - a)\left(x - \frac{a+b}{2}\right)}{2h^2}$$

et l'intégrale de polynôme $P_2(x)$ devient :

$$\int_a^b P_2(x) dx = f_0 \int_a^b L_0(x) dx + f_1 \int_a^b L_1(x) dx + f_2 \int_a^b L_2(x) dx$$

où

$$\begin{aligned} \int_a^b L_0(x) dx &= \frac{1}{2h^2} \int_a^b \left[\left(x - \frac{a+b}{2} \right) (x-b) \right] dx = \frac{h}{3} \\ \int_a^b L_1(x) dx &= \frac{1}{-h^2} \int_a^b [(x-a)(x-b)] dx = \frac{4h}{3} \\ \int_a^b L_2(x) dx &= \frac{1}{2h^2} \int_a^b \left[(x-a) \left(x - \frac{a+b}{2} \right) \right] dx = \frac{h}{3} \end{aligned}$$

avec $h = \frac{b-a}{2}$

on remplace ces équations dans l'équation d'intégrale,

$$\begin{aligned} \int_a^b P_2(x) dx &= f_0 \int_a^b L_0(x) dx + f_1 \int_a^b L_1(x) dx + f_2 \int_a^b L_2(x) dx \\ &= \frac{h}{3} f_0 + \frac{4h}{3} f_1 + \frac{h}{3} f_2 \\ &= \frac{h}{3} (f_0 + 4f_1 + f_2) \end{aligned}$$

donc :

$$\int_a^b f(x) dx = \int_a^b P_2(x) dx + \int_a^b \varepsilon_2(x) dx = \frac{h}{3} (f_0 + 4f_1 + f_2) + E_S \quad (3.10)$$

cette formule s'appelle formule de Simpson.

3.3.2 Formule généralisée de Simpson

On divisons l'intervalle $[a, b]$ en n intervalles égaux : $[x_0, x_1] = [x_1, x_2] = \dots = [x_{n-1}, x_n]$

en posant $n = 2k$, $x_0 = a$, $x_1 = a + h$, $x_n = x_{n-1} + nh$ avec $h = \frac{b-a}{n} = \frac{b-a}{2k}$

on obtient :

$$\begin{aligned}
\int_a^b f(x) dx &= \int_{x_0}^{x_2} f(x) dx + \int_{x_2}^{x_4} f(x) dx + \cdots + \int_{x_{n-2}}^{x_n} f(x) dx \\
&= \int_{x_0}^{x_2} P_2(x) dx + \int_{x_2}^{x_4} P_2(x) dx + \cdots + \int_{x_{n-2}}^{x_n} P_2(x) dx + E_s \\
&= \frac{h}{3} \left(f_0 + 4 \sum_{i=0}^{\frac{n}{2}-1} f_{2i+1} + 2 \sum_{i=1}^{\frac{n}{2}-1} f_{2i} + f_n \right) + E_s \tag{3.11}
\end{aligned}$$

Alors, la formule d'approximation de Simpson est :

$$\int_a^b f(x) dx = \frac{h}{3} \left[f_0 + 4 \sum_{i \text{ impaire}} f_i + 2 \sum_{i \text{ paire}} f_i + f_n \right] + E_s \tag{3.12}$$

3.3.3 L'erreur de la méthode de Simpson

Si f possède une dérivée quatrième continue alors en posant $M = \max_{x \in [a,b]} |f^{(4)}(x)|$, on a

$$\left| \int_a^b f(t) dt - \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] \right| \leq \frac{(b-a)^5}{2880} M$$

Démonstration

Posons $E = \int_a^b [f(t) - P_2(t)] dt$, de sorte que dans la formule ci-dessus, le premier membre est $|E|$

posons $m = \frac{a+b}{2}$ et considérons le polynôme $S(x) = P(x) + \frac{4}{(b-a)^2} [P'(m) - f'(m)] Q(x)$

où $Q(x) = (x-a)(x-m)(x-b)$.

En a, m et b le polynôme S prend les mêmes valeurs que f . De plus, on a

$$Q(m) = (m-a)(m-b) = \left(\frac{b-a}{2}\right) \left(\frac{a-b}{2}\right) = -\frac{(b-a)^2}{4}$$

donc $S'(m) = f'(m)$, le polynôme Q est de degré 3 et s'annule en a, m et b donc :

$$\int_a^b Q(t) dt = 0$$

d'après la formule d'intégration exacte, il s'ensuit que P et S ont la même intégrale sur $[a, b]$, autrement dit :

$$E = \int_a^b [f(t) - S(t)] dt$$

Posons $q(x) = (x - a)(x - m)^2(x - b)$ et écrivons l'égalité précédente sous la forme :

$$E = \int_a^b q(t) \frac{f(t) - S(t)}{q(t)} dt \quad (3.13)$$

Dans l'intégrale, le quotient n'est a priori pas défini en a, b et m mais la fonction $f - S$ est dérivable et vaut 0 en a

donc $\frac{f(t) - S(t)}{t - a}$ tend vers la limite $f'(a) - S'(a)$ quand t tend vers a , en donnant cette limite comme valeur en a au quotient, on obtient une fonction continue en a .

De même, la fonction $\frac{f(t) - S(t)}{t - a}$ se prolonge par continuité en b . Au point m , le développement limité d'ordre deux de la fonction $U = f - S$ s'écrit :

$$\begin{aligned} U(t) &= U(m) + U'(m)(t - m) + \frac{U''(m)}{2}(t - m)^2 + o[(t - m)^2] \\ &= \frac{U''(m)}{2}(t - m)^2 + o[(t - m)^2] \end{aligned}$$

car $U(m) = U'(m) = 0$. On voit que $\frac{U(t)}{(t - m)^2}$ tend vers $\left(\frac{1}{2}\right)U''(m)$ quand t tend vers m , ce qui permet encore de prolonger par continuité la fonction $\frac{U(t)}{(t - m)^2}$ en m .

Finalement, la fonction $\frac{f(t) - S(t)}{q(t)}$ est continue en tout point de $[a, b]$

Les valeurs $q(t)$ étant négatives ou nulles, nous pouvons appliquer à l'intégrale (3.18) la proposition suivante.

Proposition

Soit f une fonction continue sur $[a, b]$ et soit w une fonction intégrable à valeurs positives ou nulles et d'intégrale strictement positive alors il existe un nombre $c \in [a, b]$ tel que

$$\int_a^b w(t) f(t) dt = f(c) \int_a^b w(t) dt$$

en choisissant $w(t) = -q(t)$ on obtient :

$$E = \frac{f(c) - S(c)}{q(c)} \int_a^b q(t) dt \quad (3.14)$$

où c est un nombre dans $[a, b]$.

En raisonnant comme dans le calcul de l'erreur d'interpolation, on montre que

$$f(c) - S(c) = \frac{f^{(4)}(d)}{4!} q(c) \text{ pour certain nombre } d \in [a, b].$$

D'autre part, en intégrant le polynôme $q(t)$

on a :

$$\int_a^b q(t) dt = -\frac{(b-a)^5}{120}$$

et en reportant dans (3.14), il vient

$$E = -\frac{f^{(4)}(d)}{4!} \frac{(b-a)^5}{12} \quad (3.15)$$

3.3.4 Exercice d'application de la méthode de Simpson

Exercice :

On considère l'intégrale

$$I = \int_1^2 \frac{1}{x^2} dx$$

1. Calculer la valeur exacte de I .
2. Évaluer numériquement cette intégrale par la méthode de Simpson avec $n = 5$ sous-intervalles.

Solution :

1. La valeur exacte est $I = [-1/x]_{x=1}^{x=2} = -1/2 + 1 = 1/2$.
2. La méthode de Simpson composite à $(n + 1)$ points pour calculer l'intégrale d'une fonction f sur l'intervalle $[a, b]$ s'écrit

$$\int_a^b f(t) dt = \frac{h}{3} (f(a) + 4 \sum_{\substack{i=1 \\ i: \text{impair}}}^{n-1} f(a+ih) + 2 \sum_{\substack{i=1 \\ i: \text{pair}}}^{n-1} f(a+ih) + f(b)) \quad \text{avec } h = \frac{b-a}{n}.$$

Ici on a $f(x) = \frac{1}{x^2}$, $a = 1$, $b = 2$, $n = 5$ d'où $h = \frac{1}{5}$ et on obtient

$$\begin{aligned} I &\simeq \frac{1}{15} (f(1) + 4f(1 + 1/5) + 2f(1 + 2/5) + 4f(1 + 3/5) + 2f(1 + 4/5) + f(2)) \\ &= \frac{1}{15} (1 + 4 * 25/36 + 2 * 25/49 + 4 * 25/64 + 2 * 25/81 + 1/4) = 0.48. \end{aligned}$$

3.3.5 Programme en Matlab de la méthode de Simpson

```

        clc
        clear all
        close all
        f = input('les valeurs de la fonction pour chaque abscice')
        a = input('la valeur minimale d'intervalles')
        b = input('la valeur maximale d'intervalles')
        n = length(f) - 1
        h = (b - a)/n;
        for i = 1 : n + 1
            x(i) = a + (i - 1) * h
        end
        s1 = 0;
        for i = 2 : 2 : n - 1
            s1 = s1 + f(i);
        end
        s2 = 0;
        for i = 3 : 2 : n - 2
            s2 = s2 + f(i);
        end
        Isimpson = (h/3) * (f(1) + 4 * s1 + 2 * s2 + f(n))

```

3.4 Méthode de Newton

3.4.1 Principe de la méthode

Cette méthode regroupe quatre points de la courbe de la fonction f telle que :

$$x_0 = a, x_1 = x_0 + h, x_2 = x_0 + 2h, x_3 = x_0 + 3h \text{ et } h = \frac{b - a}{3}$$

$$\text{alors : } x_0 = a, x_1 = \frac{b + 3a}{3}, x_2 = \frac{a + 2b}{3}, x_3 = b.$$

On va utiliser dans cette méthode l'interpolation de Lagrange c'est-à-dire on remplace la fonction f sur chaque intervalle $[x_i, x_{i+1}]$ par un polynôme de degré trois, puis on calcule l'intégrale de f .

Le polynôme d'interpolation de Lagrange de degré trois s'écrit :

$$P_3(x) = \sum_{i=0}^3 L_i(x) f_i = L_0(x) f_0 + L_1(x) f_1 + L_2(x) f_2 + L_3(x) f_3$$

$$\int_a^b f(x) dx = \int_a^b P_3(x) = f_0 \int_a^b L_0(x) dx + f_1 \int_a^b L_1(x) dx + f_2 \int_a^b L_2(x) dx + f_3 \int_a^b L_3(x) dx + \int_a^b \varepsilon_3(x) dx$$

où

$$\begin{aligned} \int_a^b L_0(x) &= \frac{1}{6h^3} \int_a^b \left[\left(x - \frac{b+2a}{3} \right) \left(x - \frac{a+2b}{3} \right) (x-b) \right] dx = \frac{3h}{8} \\ \int_a^b L_1(x) &= \frac{1}{2h^3} \int_a^b \left[(x-a) \left(x - \frac{a+2b}{3} \right) (x-b) \right] dx = \frac{9h}{8} \\ \int_a^b L_2(x) &= \frac{1}{-2h^3} \int_a^b \left[(x-a) \left(x - \frac{b+2a}{3} \right) (x-b) \right] dx = \frac{9h}{8} \\ \int_a^b L_3(x) &= \frac{1}{6h^3} \int_a^b \left[(x-a) \left(x - \frac{b+2a}{3} \right) (x-b) \right] dx = \frac{3h}{8} \end{aligned}$$

alors

$$\begin{aligned} \int_a^b f(x) dx &= \int_a^b P_3(x) dx + \int_a^b \varepsilon_3(x) dx \\ &= \frac{3}{8} h [f_0 + 3f_1 + 3f_2 + f_3] + \int_a^b E_N. \end{aligned} \quad (3.16)$$

3.4.2 Formule généralisée de Newton

On divisons l'intervalle $[a, b]$ en n intervalles égaux $[x_0, x_3] = [x_3, x_6] = \dots = [x_{n-3}, x_n]$ et on considère les erreurs entre les intervalles les aussi, avec : $x_i = x_0 + ih$, $x_0 = a$, $h = \frac{b-a}{3}$

$$\begin{aligned} \int_a^b f(x) dx &= \int_{x_0}^{x_3} P_3(x) dx + \int_{x_3}^{x_6} P_3(x) dx + \dots + \int_{x_{n-3}}^{x_n} P_3(x) dx + E_N \\ &= \frac{3h}{8} \left[f_0 + 2 \sum_{i=3k}^n f_i + 3 \sum_{\substack{i=1 \\ i \neq 3k}}^{n-1} f_i + f_n \right] + E_N. \end{aligned} \quad (3.17)$$

3.5 Méthode de Newton-Cotes

Dans la méthode de Newton-Cotes de rang n , qu'on désignera dans la suite par NC_n , on prend $n_i = n$ pour tout i , et les points $\xi_{i,j}$ $0 \leq j \leq n$, sont les points équidistants

$$\xi_{i,j} = x_i + j \frac{x_{i+1} - x_i}{n}$$

divisant $[x_i, x_{i+1}]$ en l sous-intervalles égaux. Pour déterminer la formule de quadrature élémentaire, on se ramène par changement de variable à l'intervalle $[x_i, x_{i+1}] = [-1, 1]$,

subdivisé par les points $\tau_j = -1 + j\frac{2}{n}$. Le polynôme d'interpolation d'une fonction $f \in C([-1, 1])$ est donné par

$$P_n(x) = \sum_{j=0}^n f(\tau_j) L_j(x)$$

avec $L_j(x) = \frac{x - \tau_k}{k \neq j \tau_j - \tau_k}$ on a donc

$$\int_{-1}^1 f(x) dx = \int_{-1}^1 P_n(x) dx = 2 \sum_{j=0}^n w_j f(\tau_j)$$

avec $w_j = \frac{1}{2} \int_{-1}^1 L_j(x) dx$. Par suite de la symétrie des points τ_j autour de 0, on a

$$\tau_{n-j} = -\tau_j, \quad L_{n-j}(x) = L_j(-x), \quad w_{n-j} = w_j$$

pour $l = 2$ par exemple, il vient

$$\tau_0 = -1, \quad \tau_1 = 0, \quad \tau_2 = 1, \quad L_1(x) = 1 - x^2 \quad w_1 = \frac{1}{2} \int_{-1}^1 (1 - x^2) dx = \frac{2}{3}$$

d'où $w_0 = w_2 = \frac{1}{2}(1 - w_1) = \frac{1}{6}$. Après changement de variable, les coefficients w_j restent inchangés, donc on obtient les formules

$$\int_{x_i}^{x_{i+1}} f(x) dx \simeq (x_{i+1} - x_i) \sum_{j=0}^l w_j f(\xi_{i,j})$$

$$\int_a^b f(x) dx \simeq \sum_{i=0}^{k-1} (x_{i+1} - x_i) \sum_{j=0}^l w_j f(\xi_{i,j})$$

Si $f \in P_n$, alors $P_n = f$, donc la méthode de Newton-Cotes de rang n est d'ordre $\geq n$. De plus, lorsque $f \in C([-1, 1])$ est un polynôme impaire, on a

$$\int_{-1}^1 f(x) dx = 0 = 2 \sum_{j=0}^l w_j f(\tau_j)$$

Si n est paire, les formules sont donc encore exactes pour $f(x) = x^{n+1}$, et plus généralement pour $f \in P_{n+1}$ par linéarité.

On démontre en fait le résultat suivant que nous admettrons :

Proposition

Si n est pair, l'ordre de NC_n est $n + 1$

Si n est impair, l'ordre de NC_n est n

Les valeurs de w_j pour les degrés $n \leq 7$ sont

Méthode	n	w_0	w_1	w_2	w_3	w_4	w_5	w_6	(NC_n)	$E_{loc}(h)$	$E_{glob}(h)$
trapèzes	1	$\frac{1}{2}$	$\frac{1}{2}$						$1 \sim$	$f^{(2)}h^3 \sim$	$(b-a)f^{(2)}h^2$
Simpson	2	$\frac{1}{6}$	$\frac{4}{6}$	$\frac{1}{6}$					$3 \sim$	$f^{(4)}h^5 \sim$	$(b-a)f^{(4)}h^4$
Newton	3	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$				$3 \sim$	$f^{(4)}h^5 \sim$	$(b-a)f^{(4)}h^4$
Boole– Vilarceau	4	$\frac{7}{90}$	$\frac{16}{45}$	$\frac{2}{15}$	$\frac{16}{45}$	$\frac{7}{90}$			$5 \sim$	$f^{(6)}h^7 \sim$	$(b-a)f^{(6)}h^6$
Weddle– Hardy	6	$\frac{41}{840}$	$\frac{9}{35}$	$\frac{9}{280}$	$\frac{34}{105}$	$\frac{9}{280}$	$\frac{9}{35}$	$\frac{41}{840}$	$7 \sim$	$f^{(8)}h^9 \sim$	$(b-a)f^{(8)}h^8$

Pour $n \geq 8$, il apparaît des coefficients $w_j < 0$, ce qui a pour effet de rendre les formules beaucoup plus sensibles aux erreurs d'arrondis. Les méthodes NC_n ne sont donc utilisées en pratique que dans les cas ci-dessus.

Remarque

Les formules des trapèzes, de Simpson et de Newton sont des cas particulières des formules de Newton-Cotes.

3.5.1 Exercice d'application de la méthode de Newton

Exercice :

On considère l'intégrale

$$I = \int_1^2 \frac{1}{x^2} dx$$

Évaluer numériquement cette intégrale par la méthode de Newton avec $n = 5$ sous-intervalles.

Solution :

La méthode de Newton composite à $(n+1)$ points pour calculer l'intégrale d'une fonction f sur l'intervalle $[a, b]$ s'écrit

$$\int_a^b f(t) dt = \frac{3h}{8} (f(a) + 2 \sum_{i=1, i \neq 3k}^{n-1} f(a+ih) + 3 \sum_{i=1, i=3k}^{n-1} f(a+ih) + f(b)) \quad \text{avec } h = \frac{b-a}{n}.$$

Ici on a $f(x) = \frac{1}{x^2}$, $a = 1$, $b = 2$, $n = 5$ d'où $h = \frac{1}{5}$ et on obtient $x_0 = 1$, $x_1 = \frac{6}{5}$, $x_2 = \frac{7}{5}$, $x_3 = \frac{8}{5}$, $x_4 = \frac{9}{5}$, $x_5 = 2$, et

$$\begin{aligned} I &\simeq \frac{1}{8} (f(1) + 3 * f(1 + 1/5) + 3 * f(1 + 2/5) + 2 * f(1 + 3/5) + 3 * f(1 + 4/5) + f(2)) \\ &= \frac{3}{40} (1 + 3 * 25/36 + 3 * 25/49 + 2 * 25/64 + 3 * 25/81 + 1/4) = 0.492834. \end{aligned}$$

3.5.2 Programme en Matlab de la méthode de Newton

```

       clc
        clear all
        f = input('les valeurs de la fonction pour chaque abscice')
        a = input('la valeur minimale d'interval')
        b = input('la valeur maximale d'interval')
        n = length(f) - 1
        h = (b - a)/n;
        for i = 1 : n + 1
            x(i) = a + (i - 1) * h
        end
        s1 = 0;
        for i = 4 : 3 : n - 1
            s1 = s1 + f(i);
        end
        s2 = 0;
        for i = 1 : n - 1
            s2 = s2 + f(i);
        end
        Inewton = (3 * h/8) * (f(1) - s1 + 3 * s2 + f(n))

```

3.6 Méthode de Gauss

La méthode de Gauss est une méthode qui permet de calcul l'intégrale par une somme pondéré prise en un certain nombre de points du domaine d'intégration.

Cette méthode est une méthode de quadrature pour un polynôme de degré $2n - 1$ avec n points pris sur le domaine d'intégration.

Si ce dernier est $]a, b[$ (pas nécessairement ouvert), la méthode est de la forme :

$$\forall P \in R_{2n-1}, \quad \int_a^b P(x) \pi(x) dx = \sum_{k=1}^n \lambda_k P(x_k) \quad (3.18)$$

où

$\pi(x) :]a, b[\rightarrow \mathbb{R}_+$ est une fonction de pondération.

λ_k : sont appelé les coefficients de quadrature.

x_k : sont des réels distincts unique et sont les racines de polynômes orthogonaux.

Dans ce qui suit $(P_n)_{n \in \mathbb{N}}$ est une suite de polynômes orthogonaux associé à la fonction de pondération avec P_n de degré n pour tout $n \in \mathbb{N}$.

Pour tout entier naturel n , on note :

$$P_n(x) = \sum_{k=0}^n \alpha_k^{(n)} x^k$$

avec $\alpha_n^{(n)} > 0$

la suite $(P_n)_{n \in \mathbb{N}}$ est aussi définie par la relation de récurrence :

$$\begin{cases} P_{-1}(x) = 0 & , & P_0(x) = \alpha_0^{(0)} = \frac{1}{\sqrt{\int_a^b \pi(x) dx}} \\ b_{n+1}P_{n+1}(x) + a_nP_n(x) + b_nP_{n-1}(x) = xP_n(x) & n \geq 0 \end{cases} \quad (3.19)$$

avec

$$\begin{cases} a_0 = -\frac{\alpha_0^{(1)}}{\alpha_1^{(1)}} & b_0 = 0 \\ a_n = \frac{\alpha_{n-1}^{(n)}}{\alpha_n^{(n)}} - \frac{\alpha_n^{(n+1)}}{\alpha_{n+1}^{(n+1)}} & b_n = \frac{\alpha_{n-1}^{(n-1)}}{\alpha_n^{(n)}} \quad n \geq 1 \end{cases}$$

pour tout $n \geq 1$, le polynôme P_n a n racines distinctes dans $I =]a, b[$.

3.6.1 Comment calculer l'intégrale

Pour calculer l'intégrale, il faut déterminer les réels x_k et les coefficients λ_k .

Lemme 01 :

Pour $n \in \mathbb{N}$, il existe des coefficients réels λ_k ($1 \leq k \leq n$) tels que la condition (3.18) soit vérifié si et seulement si les réels x_k ($1 \leq k \leq n$) sont les racines du polynôme P_n .

Démonstration

Supposons qu'il existe des coefficients λ_k ($1 \leq k \leq n$) tels que la condition (3.18) soit vérifié.

On désigne par W_n le polynôme unitaire de degré n qui s'annule en chacun des x_k ($1 \leq k \leq n$).

Soit $W_n = \prod_{k=1}^n (x - x_k)$

Pour tout polynôme $Q \in R_{n-1}[x]$, on a $W_n Q \in R_{2n-1}[x]$

et

$$\langle W_n | Q \rangle = \int_a^b W_n(x) Q(x) \pi(x) dx = \sum_{k=1}^n \lambda_k W_n(x_k) Q(x_k) = 0$$

c'est à dire que W_n est orthogonal à $R_{n-1}[x]$, ce polynôme de degré n est donc proportionnel à P_n , et les x_k ($1 \leq k \leq n$) sont des racines de P_n .

Réciproquement

Supposons que les x_k ($1 \leq k \leq n$) soit les racines de P_n .

Par division euclidienne, tout polynôme $P \in R_{2n-1}[x]$ s'écrit sous la forme

$$P = QP_n + R$$

avec Q, R dans $R_{n-1}[x]$ et on a :

$$\int_a^b P(x) \pi(x) dx = \langle P_n | Q \rangle + \int_a^b R(x) \pi(x) dx = \int_a^b R(x) \pi(x) dx$$

puisque $P_n \in (R_{n-1}[x])^\perp$, en remarquant que :

$P(x_k) = R(x_k)$ pour tout k compris entre 1 et n

On déduit qu'il nous suffit de montrer que (3.18) est vérifié sur $R_{n-1}[x]$ ce qui équivaut à prouver l'existence des coefficients λ_k ($1 \leq k \leq n$) solution du système linéaire de n équation à n inconnues :

$$\sum_{i=1}^n (x_i)^{k-1} \lambda_i = \int_a^b x^{k-1} \pi(x) dx \quad (1 \leq k \leq n)$$

le déterminant de ce système étant $\prod_{1 \leq j < i \leq n} (x_i - x_j) \neq 0$ on est assuré de l'existence et de l'unicité d'une solution λ_k les coefficients λ_k ($1 \leq k \leq n$) qu'est définie sont appelé coefficients de Christoffel associés à la fonction poids π et à l'intervalle I .

On suppose dans ce qui suit que pour $n \geq 1$, les réels x_k sont les racines du polynôme P_n et on désigne par $(L_k)_{1 \leq k \leq n}$ la base de Lagrange de $R_{n-1}[x]$ définie par :

$\forall k \in \{1 \dots n\}$

$$\left\{ \begin{array}{l} L_k \in R_{n-1}[x] \\ L_k(x_i) = \begin{cases} 0 & \text{si } i \neq k \\ 1 & \text{si } i = k \end{cases} \quad 1 \leq i \leq n \end{array} \right.$$

cette base est également définie par :

$$L_k(x) = \prod_{\substack{i=1 \\ i \neq k}}^n (x - x_i) x_k - x_i = \frac{P_n(x)}{(x - x_k) P'_n(x_k)} \quad (1 \leq k \leq n) \quad (3.20)$$

le résultat qui suit nous donne une formule explicite des coefficient de Christoffel.

Lemme 2 :

Pour tout k compris entre 1 et n on a

$$\lambda_k = \frac{1}{b_n} \frac{1}{P'_n(x_k) P_{n-1}(x_k)}$$

Démonstration

Pour tout k compris entre 1 et n , on a :

$$\int_a^b L_k(x) \pi(x) dx = \sum_{i=1}^n \lambda_i L_k(x_i) = \lambda_k$$

soit :

$$\lambda_k = \frac{1}{P'_n(x_k)} \int_a^b \frac{P_n(x)}{x - x_k} \pi(x) dx \quad (3.21)$$

en utilisant l'identité de Darboux-Christoffel [4], on a :

$$(x - x_k) \sum_{i=1}^n P_i(x) P_i(x_k) = -b_{n+1} P_n(x) P_{n+1}(x_k)$$

ce qui donne, pour $x \neq x_k$:

$$\frac{P_n(x)}{x - x_k} = -\frac{1}{b_{n+1}} \frac{1}{P_{n+1}(x_k)} \sum_{i=0}^n P_i(x) P_i(x_k)$$

on remplace cette équation dans l'équation(3.21), on obtient

$$\lambda_k = -\frac{1}{b_{n+1}} \frac{1}{P_{n+1}(x_k) P'_n(x_k)} \sum_{i=0}^n \frac{P_i(x_k)}{\alpha_0^{(0)}} \langle P_i | P_0 \rangle$$

tenant compte de l'orthogonalité de la famille $(P_i)_{0 \leq i \leq n}$ on en déduit que :

$$\lambda_k = -\frac{1}{b_{n+1}} \frac{1}{P_{n+1}(x_k) P'_n(x_k)}$$

d'autre part, en utilisant la relation de récurrence(3.19) on a :

$$b_{n+1} P_{n+1}(x_k) + b_n P_{n-1}(x_k) = 0$$

soit :

$$b_{n+1} = -b_n \frac{P_{n-1}(x_k)}{P_{n+1}(x_k)}$$

et

$$\lambda_k = \frac{1}{b_n} \frac{1}{P'_n(x_k) P_{n-1}(x_k)}$$

3.6.2 Généralisation pour un intervalle fermé

Le domaine d'intégration $[a, b]$ doit être changé (changement de variable) en $[-1, 1]$ avant d'appliquer les méthodes de Gauss.

Le changement se déroule ainsi :

$$\int_a^b f(t) dt = \frac{b-a}{2} \int_{-1}^1 f\left(\frac{b-a}{2}x + \frac{a+b}{2}\right) dx = \frac{b-a}{2} \int_{-1}^1 g(x) dx \quad (3.22)$$

On peut écrire l'approximation de la valeur de l'intégrale comme suit :

$$\int_{-1}^1 g(x) dx = \sum_{i=1}^n \lambda_k g(x_k) + E$$

tel que E est l'erreur de la méthode de Gauss.

3.6.3 Les différents types de la méthode de Gauss

Pour chaque forme d'intégrale il existe une famille de polynômes orthogonaux, donc des points d'interpolation $(x_i, i = 1, \dots, n)$ et des coefficients $(\lambda_i, i = 1, \dots, n)$ et des fonctions de poids qui permet de calculer l'intégrale.

Méthode de Gauss-Tchebycheff

la fonction de poids est : $\pi(x) = \frac{1}{\sqrt{1-x^2}}$

l'intervalle considéré est : $] -1, 1[$

les polynômes orthogonaux sont :

$$\begin{cases} T_0(x) = 1 \\ T_1(x) = x \\ \vdots \\ T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) = \cos(n \arccos(x)) \end{cases} \quad (3.23)$$

cette méthode est adaptée au calcul des intégrales de la forme :

$$\int_{-1}^1 \frac{f(x)}{\sqrt{1-x^2}} dx = \sum_{i=1}^n \lambda_i f(x_i) + E \quad (3.24)$$

Méthode de Gauss-Laguerre

La fonction de poids est : $\pi(x) = e^{-x}$

l'intervalle considéré est : $] 0, +\infty[$

les polynômes orthogonaux sont :

$$\left\{ \begin{array}{l} L_0(x) = 1 \\ L_1(x) = 1 - x \\ \vdots \\ L_n(x) = \frac{2n-1-x}{n}L_{n-1}(x) - \frac{n-1}{n}L_{n-2}(x) \end{array} \right. \quad (3.25)$$

cette méthode est adaptée au calcul des intégrales de la forme :

$$\int_0^{+\infty} e^{-x} f(x) dx = \sum_{i=1}^n \lambda_i f(x_i) + E \quad (3.26)$$

Méthode de Gauss-Legendre

La fonction de poids est : $\pi(x) = 1$

l'intervalle considéré est : $] -1, 1[$

les polynômes orthogonaux sont :

$$\left\{ \begin{array}{l} P_0(x) = 1 \\ P_1(x) = x \\ \vdots \\ P_{n+1}(x) = \frac{1}{n+1} ((2n+1)x P_n(x) - n P_{n-1}(x)) \forall n > 0 \end{array} \right. \quad (3.27)$$

cette méthode est adaptée au calcul des intégrales de la forme :

$$\int_{-1}^1 f(x) dx = \sum_{i=1}^n \lambda_i f(x_i) + E \quad (3.28)$$

Méthode de Gauss-Hermite

La fonction de poids est : $\pi(x) = e^{-x^2}$

l'intervalle considéré est : $] -\infty, +\infty[$

les polynômes orthogonaux sont :

$$\left\{ \begin{array}{l} H_0(x) = 1 \\ H_1(x) = x \\ \vdots \\ H_{n+1}(x) = xH_n(x) - nH_{n-1}(x) \end{array} \right. \quad (3.29)$$

cette méthode est adaptée au calcul des intégrales de la forme :

$$\int_a^b e^{-x^2} f(x) dx = \sum_{i=1}^n \lambda_i f(x_i) + E \quad (3.30)$$

3.7 Majoration de l'erreur de Gauss

On se donne une fonction f dans $C(]a, b[)$ tel que l'intégrale $\int_a^b f(t)\pi(t)dt$ soit convergente, et pour tout entier naturel non nul n , on note :

$$E_n(f) = \int_a^b f(t)\pi(t)dt - \sum_{k=1}^n \lambda_k f(x_k)$$

Cette expression est l'erreur de Gauss

Théorème :

Si $f \in C^{2n}(]a, b[)$ est telle que l'intégrale $\int_a^b f(t)\pi(t)dt$ est convergente et $f^{(2n)}$ est bornée sur $]a, b[$ alors :

$$|E_n(f)| \leq \frac{\|f^{(2n)}\|_\infty}{(2n)!(\alpha_n^{(n)})^2} = \frac{\sup_{x \in]a, b[} |f^{(2n)}(x)|}{(2n)!(\alpha_n^{(n)})^2} \quad (3.31)$$

Démonstration :

On désigne par H_{2n-1} le polynôme d'interpolation d'Hermite définie par :

$$\begin{cases} H_{2n-1} \in R_{2n-1}[x] \\ H_{2n-1}(x_k) = f(x_k), \quad H'_{2n-1}(x_k) = f'(x_k), \quad 1 \leq k \leq n \end{cases}$$

en considérant que

$$\int_a^b H_{2n-1}(t) \pi(t) dt = \sum_{k=1}^n \lambda_k H_{2n-1}(x_k) = \sum_{k=1}^n \lambda_k f(x_k)$$

on déduit que l'erreur de quadrature dans la méthode de Gauss est donnée par :

$$E_n(f) = \int_a^b f(t) \pi(t) dt - \sum_{k=1}^n \lambda_k H_{2n-1}(x_k) = \int_a^b (f(t) - H_{2n-1}(t))\pi(t)dt \quad (3.32)$$

d'autre part, on sait que pour $f \in C^{2n}(]a, b[)$, l'erreur d'interpolation d'Hermite est de la forme :

$$f(x) - H_{2n-1}(x) = \frac{f^{(2n)}(c_x)}{(2n)!(\alpha_n^{(n)})^2} (P_n(x))^2$$

où $c_x \in]a, b[$ et pour $f^{(2n)}$ bornée sur $]a, b[$, on en déduit que :

$$|E_n(f)| \leq \frac{\|f^{(2n)}\|_\infty}{(2n!(\alpha_n^{(n)})^2)} \|P_n\|^2 = \frac{\sup_{x \in]a,b[} |f^{(2n)}(x)|}{(2n!(\alpha_n^{(n)})^2)}$$

3.7.1 Exercice d'application de la méthode de Gauss

Exercice :

En utilisant la méthode de Gauss ($n = 2$), trouver la valeur approchée de $\int_0^{\pi/2} \sin(x) dx$.

Solution :

On a $\int_a^b f(y) dy = \frac{b-a}{2} \sum_{i=1}^n A_i f(y_i)$, où $y_i = \frac{b+a}{2} + \frac{b-a}{2} x_i$ et les x_i sont les racines du

polynôme de Legendre.

Pour $n = 2$ on a : $x_1 = 0.5773$, $x_2 = -0.5773$, $A_1 = A_2 = 1$.

$$\int_0^{\pi/2} f(y) dy = \frac{\pi/2-0}{2} (A_1 f(y_1) + A_2 f(y_2)) \text{ où } y_1 = \frac{\pi}{4} + \frac{\pi}{4} x_1 \text{ et } y_2 = \frac{\pi}{4} + \frac{\pi}{4} x_2$$

$$y_1 = \frac{\pi}{4} + \frac{\pi}{4}(0.5773) = 1.2388 \text{ d'où } f(y_1) = 0.9454$$

$$y_2 = \frac{\pi}{4} + \frac{\pi}{4}(-0.5773) = 0.3320 \text{ d'où } f(y_2) = 0.3259$$

donc

$$\int_0^{\pi/2} \sin(y) dy = \frac{\pi}{4} (0.9454 + 0.3259) = 0.9985$$

3.7.2 Programme en Matlab de la méthode de Gauss

```

clc
clear all
close all

t = input('les valeurs des abscises')
w = input('les valeurs de w')
a = input('la valeur minimale d'interval')
b = input('la valeur maximale d'interval')
n = length(t)
for i = 1 : n
    f(i) = t(i)^2 - t(i) + 2.
end
s = 0;
for i = 1 : n
    s = s + w(i) * f(i);
end
Igauss = ((b - a)/2) * s

```

3.8 Série des exercices

Exercice 3 - 1 :

En utilisant la table ci-dessous, calculer $\int_a^b f(x)dx$ par :

- 1- la méthode du trapèze
- 2- par la méthode de Simpson

x	1.1	1.3	1.5
$f(x)$	3.0042	3.6693	4.4817

Exercice 3 - 2 :

Soient les données suivantes :

x	1.0	1.2	1.4	1.6	1.8
$f(x)$	1.543	1.811	2.151	2.577	3.107

Trouver la valeur approchée de $\int_{1.0}^{1.8} f(x)dx$, en utilisant :

- 1- la méthode du trapèze généralisée.
- 2- la méthode de Simpson généralisée.

Exercice 3 - 3 :

On définit la fonction $g(x) = \frac{1}{1-x}$

a- Calculer $F(x) = \int_0^x g(t) dt$, $x < 1$, et quelle est la valeur de $F(x)$ en $x = 2/3$?

b- Donner le degré et l'expression du polynôme d'interpolation de Lagrange qui interpole $g(x)$ aux points $0, 1/3$ et $2/3$.

c- Trouver des coefficients c_0, c_1 et c_2 tels que pour tout polynôme P de degré inférieur ou égal à 2, on ait $\int_0^2 P(x) dx = c_0P(0) + c_1P(1) + c_2P(2)$.

d- En utilisant un changement de variable, déduire de la formule précédente les coefficients d_0, d_1 et d_2 tels que pour tout polynôme Q de degré inférieur ou égal à 2, on ait

$$\int_0^{2/3} Q(x) dx = d_0Q(0) + d_1Q\left(\frac{1}{3}\right) + d_2Q\left(\frac{2}{3}\right).$$

Utiliser cette formule pour donner une valeur approchée de $\ln(3)$.

Chapitre 4

La résolution du système des équations linéaires

4.1 Introduction

Dans ce chapitre, nous rappelons les notions élémentaires d'algèbre linéaire que nous utiliserons dans le reste de l'ouvrage. Pour les démonstrations et pour plus de détails.

4.2 Généralités sur les matrices

Nous allons passer en revue certaines définitions sur les matrices qui interviennent en algèbre matricielle.

4.2.1 Matrice inverse

S'il existe, inverse de A la matrice notée A^{-1} telle que $A A^{-1} = A^{-1} A = I$.
Une matrice qui possède un inverse s'appelle inversible ou régulière.

4.2.2 Matrice diagonales et triangulaires

* Une matrice A telle que $a_{ij} = 0$ si $i \neq j$ s'appelle une matrice diagonale. Son déterminant vaut $\prod_{i=1}^n a_{ii}$. Son inverse est la matrice diagonale dont les éléments diagonaux sont $1/a_{ii}$. A^{-1} n'existe donc que si $a_{ii} \neq 0$ pour $i = 1, 2, \dots, n$.

* Une matrice A telle que $a_{ij} = 0$ pour $i > j$ s'appelle une matrice triangulaire supérieure. Son déterminant est égal à $\prod_{i=1}^n a_{ii}$. Son inverse est une matrice triangulaire supérieure. Le produit de deux matrices supérieures est une matrice supérieure.

* Une matrice A telle que $a_{ij} = 0$ pour $i < j$ s'appelle une matrice triangulaire inférieure. Son déterminant est égal à $\prod_{i=1}^n a_{ii}$. Son inverse est une matrice triangulaire inférieure. Le produit de deux matrices inférieures est une matrice inférieure.

Matrice symétrique

Une matrice est dite symétrique si $A^T = A$. Donc $a_{ij} = a_{ji} \forall i, j$

Il faut faire attention que le produit de deux matrices symétriques n'est pas obligatoirement une matrice symétrique.

Matrice Conjuguée

Soit A une matrice dont les éléments appartiennent à \mathbb{C} . B est la conjuguée de A si $b_{ij} = \overline{a_{ij}} \forall i, j$. On notera $B = \overline{A}$ on a $\overline{\overline{A}} = A$, $\overline{BA} = \overline{B} \overline{A}$, $\overline{(A)^{-1}} = \overline{A}^{-1}$

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

Matrice adjointe

Soit A une matrice dont les éléments appartiennent à \mathbb{C} . B est l'adjointe de A si $b_{ij} = \overline{a_{ji}} \forall i, j$.

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

Matrice hermitienne

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

Matrice orthogonale

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

Matrice unitaire

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

Matrice normale

Une matrice A est dite normale si $A^*A = A A^*$. Une matrice unitaire est normale.

4.2.3 Matrice définie positive

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

Exemple :

Soit la matrice $A = \begin{bmatrix} 1 & 1 \\ 1 & 5 \end{bmatrix}$

Soit $x = (x_1, x_2)^T$ un vecteur quelconque non nul. On a : $Ax = \begin{bmatrix} x_1 + x_2 \\ x_1 + 5x_2 \end{bmatrix}$ et $x^T Ax = x_1(x_1 + x_2) + x_2(x_1 + 5x_2) = (x_1 + x_2)^2 + 4x_2^2$ qui est toujours strictement positif quels que soient x_1 et x_2 non nuls simultanément.

Les éléments diagonaux d'une matrice définie positive sont strictement positifs (car $a_{ii} = e_i^T A e_i > 0$)

4.2.4 Matrice de permutation

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

$$\text{Soient } P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, A = \begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{bmatrix}$$

on obtient :

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

4.2.5 Les propriétés de déterminant

Soient A et B deux matrices carrées et λ est un nombre réel ou complexe

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

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

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

$$\det(B^{-1}AB) = \det(A)$$

$$\det(\lambda A) = \lambda^n \det(A)$$

Soit $A = [a_1, a_2, \dots, a_n]$ alors

$$\det[\lambda a_1, a_2, \dots, a_n] = \lambda \det[a_1, a_2, \dots, a_n]$$

$$\det[a_1, a_2 + \lambda a_1, \dots, a_n] = \det[a_1, a_2, \dots, a_n]$$

$$\det[a_2, a_1, \dots, a_n] = -\det[a_1, a_2, \dots, a_n]$$

4.3 Généralités sur les systèmes linéaires

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

$$Ax = b$$

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

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad (4.1)$$

ou encore

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases} \quad (4.2)$$

Ce système a une solution unique si et seulement si $\det(A) \neq 0$. Si ce déterminant est nul alors le système peut avoir une infinité de solutions où n'en avoir aucune. Prenons un exemple :

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

La seconde équation est le double de la première et le déterminant de la matrice est nul. Si l'on donne à x_1 une valeur quelconque, alors $x_2 = \frac{1-x_1}{2}$. Si l'on donne à x_2 une valeur quelconque, alors $x_1 = 1 - 2x_2$. Ce système à donc une infinité de solutions. Par contre si nous remplaçons la seconde équation par $2x_1 + 4x_2 = a \neq 2$, alors on ne peut pas avoir simultanément $2(x_1 + 2x_2) = 2(1) = a \neq 2$ et le système n'a donc aucune solution.

4.3.1 Cas d'une matrice diagonale

Tous les éléments sont nuls sauf ceux diagonale principale

$$A = \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ 0 & a_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & a_n \end{bmatrix}$$

La résolution $Ax = b$ est alors immédiate, si $x = (x_1, x_2, \dots, x_n)^T$ et $b = (b_1, b_2, \dots, b_n)^T$, on a $x_i = \frac{b_i}{a_i}$, $i = 1, 2, \dots, n$.

ceci es supposant bien sur que tous les a_i sont non nuls.

4.3.2 Cas d'une matrice triangulaire supérieure (où inférieure)

Tous les éléments au-dessous (où au-dessus) de la diagonale sont nuls

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix}$$

Dans ce cas on utilisera toujours une méthode directe de résolution par remontée soit :

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

On commence par résoudre le dernière équation ; on substituée le résultat obtenu pour x_n dans la précédente, ce qui permet de calculer x_{n-1} , etc...

$$\left\{ \begin{array}{l} x_n = b_n/a_{nn} \\ x_{n-1} = (b_{n-1} - a_{n-1n}x_n)/a_{n-1n-1} \\ \vdots \\ x_i = (b_i - a_{in}x_n - a_{in-1}x_{n-1} - \dots - a_{ii+1}x_{i+1})/a_{ii} \end{array} \right.$$

Le cas des matrices triangulaires inférieures se traite de façons identique par un procédé de descente.

Il existe deux grandes familles de méthodes de résolution :

a)- Les méthodes directes qui permettent de résoudre le système soit par triangularisation ou soit par factorisation de la matrice A . Les principales méthodes sont :

- La méthode de Gauss.
- La méthode de Gauss-Jordan.
- La factorisation LU .

b)- Les méthodes itératives qui introduisent une notion de convergence vers la solution. Les principales méthodes sont :

- La méthode de Jacobi.
- La méthode de Gauss-Seidel.

4.4 Les méthodes directes

4.4.1 Méthode de Gauss

La méthode d'élimination de Gauss a pour but de transformer le système de départ en un système ayant la même solution de la forme $Ux = c$ où U est une matrice triangulaire supérieure et c un vecteur. La résolution est en deux étapes :

- * Triangularisation de la matrice A .
- * Résolution du système triangulaire.

Triangularisation de la matrice A

On suppose une nouvelle matrice s'écrit comme $B = [A; b]$, on injecte le vecteur b et la matrice A d'une nouvelle matrice B .

Pour triangularisation la matrice A , on utilise la technique suivante :

Supposons pour simplifier la méthode que $a_{11} \neq 0$, a_{11} est le premier pivot de l'élimination. On commence la triangularisation, en annulant les éléments de la première colonne de A situées en dessous de la diagonale. Cette opération peut être réalisée en multipliant A , à gauche, par la matrice $P^{(1)}$

$$P^{(1)} = \begin{bmatrix} 1/a_{11} & 0 & 0 & \dots & 0 \\ -a_{21}/a_{11} & 1 & 0 & \dots & 0 \\ -a_{31}/a_{11} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_{n1}/a_{11} & 0 & 0 & \dots & 1 \end{bmatrix}$$

dont l'inverse est :

$$(P^{(1)})^{-1} = \begin{bmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ a_{21} & 1 & 0 & \cdots & 0 \\ a_{31} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & 0 & 0 & \cdots & 1 \end{bmatrix}$$

Le système $Ax = b$ est alors équivalent au système $A^{(1)}x = b^{(1)}$, telle que $A^{(1)} = P^{(1)}A$ et $b^{(1)} = P^{(1)}b$ où $B^{(1)} = [A^{(1)}; b^{(1)}]$ de la forme :

$$B^{(1)} = \begin{bmatrix} 1 & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} & a_{1n+1}^{(1)} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} & a_{2n+1}^{(1)} \\ 0 & a_{32}^{(1)} & a_{33}^{(1)} & \cdots & a_{3n}^{(1)} & a_{3n+1}^{(1)} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & a_{n2}^{(1)} & a_{n3}^{(1)} & \cdots & a_{nn}^{(1)} & a_{nn+1}^{(1)} \end{bmatrix}$$

dont les coefficients $a_{ij}^{(1)}$ sont donnés par les relations :

$$\begin{cases} a_{ij}^{(1)} = a_{ij}/a_{11} & j = 2, 3, \dots, n+1 \\ a_{ij}^{(1)} = a_{ij} - a_{i1}a_{1j}^{(1)} & 2 \leq i \leq n \text{ et } 2 \leq j \leq n+1 \end{cases}$$

comme $\det(P^{(1)}) = 1/a_{11}$, on a : $\det(A^{(1)}) = \det(A)/a_{11} \neq 0$,

La matrice $A^{(1)}$ est régulière et l'un au moins de ses éléments $a_{i2}^{(1)}$, $i = 2, 3, \dots, n$ est différent de zéro. Après d'éventuelles permutations de lignes, on peut supposer que $a_{22}^{(1)} \neq 0$ et appliquons de nouveau le même procédé à la deuxième colonne de $A^{(1)}$. On obtient de proche en proche à la $k^{\text{ième}}$ application du procédé un système équivalent au système $Ax = b$ de la forme : $A^{(k)}x = b^{(k)}$ où les matrices $A^{(k)}$ et $b^{(k)}$ sont données par : $A^{(k)} = P^{(k)}A^{(k-1)} = P^{(k)}P^{(k-1)} \dots P^{(1)}A$

$$b^{(k)} = P^{(k)}b^{(k-1)} = P^{(k)}P^{(k-1)} \dots P^{(1)}b$$

et sont de la forme :

$$B^{(k)} = \begin{bmatrix} 1 & a_{12}^{(k)} & \cdots & a_{1k}^{(k)} & a_{1k+1}^{(k)} & \cdots & a_{1n}^{(k)} & a_{1n+1}^{(k)} \\ \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & & \\ 0 & 0 & \cdots & 1 & a_{kk+1}^{(k)} & \cdots & a_{kn}^{(k)} & a_{kn+1}^{(k)} \\ 0 & 0 & \cdots & 0 & a_{k+1k+1}^{(k)} & \cdots & a_{k+1n}^{(k)} & a_{k+1n+1}^{(k)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & a_{nk+1}^{(k)} & \cdots & a_{nn}^{(k)} & a_{nn+1}^{(k)} \end{bmatrix}$$

On continue la triangularisation, lorsque $a_{k+1k+1}^{(k)}$ est non nul, en multipliant $A^{(k)}$ à gauche par la matrice :

$$P^{(k+1)} = \begin{bmatrix} 1 & 0 & \cdots & a_{1k}^{(k)} & 0 & \cdots & 0 \\ \vdots & 1 & \ddots & 0 & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & 0 & 1/a_{k+1k+1}^{(k)} & \cdots & 0 \\ 0 & 0 & \cdots & 0 & -a_{k+2k+1}^{(k)}/a_{k+1k+1}^{(k)} & \ddots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & -a_{nk+1}^{(k)}/a_{k+1k+1}^{(k)} & \cdots & 1 \end{bmatrix}$$

les éléments de la matrice $A^{(k+1)}$ et le vecteur $b^{(k+1)}$ s'écrivent dans l'algorithme suivante :

$$\begin{cases} a_{ij}^{(k+1)} = a_{ij}^{(k)} / a_{kk}^{(k)} \\ a_{ij}^{(k+1)} = a_{ij}^{(k)} - a_{ik}^{(k)} a_{kj}^{(k+1)} \\ b_i^{(k+1)} = a_{in+1}^{(k+1)} \end{cases}$$

et ainsi de suite jusqu'à $k = n - 1$. La matrice $A^{(n)} = P^{(n)}P^{(n-1)} \dots P^{(1)}A$ est triangulaire supérieure et la résolution du système $A^{(n)}x = b^{(n)}$ système équivalent à $Ax = b$ est immédiate. On a trouvé une matrice de passage $P = P^{(n)}P^{(n-1)} \dots P^{(1)}$ telle que la matrice $A^{(n)} = U = PA$ soit triangulaire supérieure et le nouveau vecteur $b^{(n)} = c = Pb$.

Après triangularisation, le système $Ux = c$ se résout en cascade en commençons par x_n .

Remarques

La méthode de Gauss n'est correctement définie que si les pivots $a_{k+1k+1}^{(k)}$ sont non nuls pour $k = 1, 2, \dots, n - 1$. Si le terme diagonal $a_{k+1k+1}^{(k)}$ est nul, il faut choisir un autre terme non nul de la colonne k pour terminer les calculs, en permutant l'ordre des lignes de la matrice.

4.4.2 Exercice d'application de la méthode de Gauss

Exercice :

On considère la matrice A et le vecteur b

$$A = \begin{bmatrix} 6 & -2 & 2 \\ -2 & 5 & 0 \\ 2 & 0 & 7 \end{bmatrix}, b = \begin{bmatrix} 0 \\ 23 \\ 16 \end{bmatrix}$$

Résoudre le système $Ax = b$ par la méthode de triangularisation de Gauss. Donner la décomposition LU de A où les matrices L et U sont triangulaires inférieure et supérieure respectivement.

Solution :

Nous utilisons l'algorithme de la méthode de Gauss :

on a à la première étape :

$$P^{(1)} = \begin{bmatrix} \frac{1}{6} & 0 & 0 \\ \frac{1}{3} & 1 & 0 \\ -\frac{1}{3} & 0 & 1 \end{bmatrix}, \quad \text{d'où } A^{(1)} = P^{(1)}A = \begin{bmatrix} 1 & -\frac{1}{3} & \frac{1}{3} \\ 0 & \frac{13}{3} & \frac{19}{3} \\ 0 & \frac{2}{3} & \frac{19}{3} \end{bmatrix} \quad \text{et } b^{(1)} = P^{(1)}b = \begin{bmatrix} 0 \\ 23 \\ 16 \end{bmatrix}$$

Puis à la deuxième étape :

$$P^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{3}{13} & 0 \\ -0 & -\frac{2}{13} & 1 \end{bmatrix}, \quad \text{d'où } A^{(2)} = P^{(2)}A^{(1)} = \begin{bmatrix} 1 & -\frac{1}{3} & \frac{1}{13} \\ 0 & 1 & \frac{2}{13} \\ 0 & 0 & \frac{81}{13} \end{bmatrix} \quad \text{et } b^{(2)} = P^{(2)}b^{(1)} = \begin{bmatrix} 0 \\ \frac{69}{13} \\ \frac{162}{13} \end{bmatrix} \quad \blacksquare$$

et en fin à la dernière étape :

$$P^{(3)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{13}{81} \end{bmatrix}, \quad \text{d'où } A^{(3)} = P^{(3)}A^{(2)} = \begin{bmatrix} 1 & -\frac{1}{3} & \frac{1}{13} \\ 0 & 1 & \frac{2}{13} \\ 0 & 0 & 1 \end{bmatrix} \quad \text{et } x = b^{(3)} = P^{(3)}b^{(2)} = \begin{bmatrix} 0 \\ \frac{69}{13} \\ 2 \end{bmatrix} \quad \blacksquare$$

Il ne reste plus qu'à résoudre le système triangulaire : $A^{(3)}x = b^{(3)}$

$$\text{ce qui donne } \begin{cases} x_3 = 2 \\ x_2 = 5 \\ x_1 = 1 \end{cases} \quad \text{et donc : } x = \begin{bmatrix} 1 \\ 5 \\ 2 \end{bmatrix}.$$

4.4.3 Programme en Matlab de la méthode de Gauss

```

        clc
        clear all
        close all
        a = input('entrer la matrice a =')
        b = input('donner le second membre du système')
        n = length(b)
        aa = [a b]
        for k = 1 : n
            for i = k + 1 : n
                m(i, k) = aa(i, k)/aa(k, k);
                for j = 1 : n + 1
                    aa(i, j) = aa(i, j) - m(i, k) * aa(k, j);
                end
            end
        end
        a = aa(:, 1 : n);
        b = aa(:, n + 1);
        x(n) = b(n)/a(n, n);
        for i = n - 1 : -1 : 1
            s = 0;
            for k = i + 1 : n
                s = s + a(i, k) * x(k);
            end
            x(i) = (b(i) - s)/a(i, i);
        end
        x
    
```

4.4.4 Factorisation LU

La factorisation LU (Lower Upper) pour une matrice A inversible consiste à déterminer deux matrices triangulaires, l'une inférieure L et l'autre supérieure U telles que $A = LU$.

Cette décomposition, unique, existe si et seulement si les n mineures de A sont non nuls.

La matrice U est obtenue par la méthode d'élimination de Gauss. Et la matrice L fait intervenir les pivots successifs de l'algorithme de Gauss.

On a $A = LU$ et on a aussi $A^{(n)} = U = P^{(n)}P^{(n-1)} \dots P^{(1)}A = L^{-1}A$

alors

$$L = (P^{(1)})^{-1} (P^{(2)})^{-1} \dots (P^{(n)})^{-1} \quad (4.3)$$

Une fois calculée L et U , résoudre le système de départ consiste à résoudre successivement les deux systèmes triangulaires

$$\begin{cases} Ly = b \\ Ux = y \end{cases}$$

4.4.5 Exercice d'application de la méthode de Factorisation LU

Exercice :

On considère la matrice A et le vecteur b

$$A = \begin{bmatrix} 6 & -2 & 2 \\ -2 & 5 & 0 \\ 2 & 0 & 7 \end{bmatrix}, b = \begin{bmatrix} 0 \\ 23 \\ 16 \end{bmatrix}$$

Résoudre le système $Ax = b$ par la méthode de triangularisation de Gauss. Donner la décomposition LU de A où les matrices L et U sont triangulaires inférieure et supérieure respectivement.

Solution :

Nous utilisons l'algorithme de la méthode de Gauss :

on a à la première étape :

$$P^{(1)} = \begin{bmatrix} \frac{1}{6} & 0 & 0 \\ \frac{1}{3} & 1 & 0 \\ -\frac{1}{3} & 0 & 1 \end{bmatrix}, \quad \text{d'où } A^{(1)} = P^{(1)}A = \begin{bmatrix} 1 & -\frac{1}{3} & \frac{1}{3} \\ 0 & \frac{13}{3} & \frac{2}{3} \\ 0 & \frac{2}{3} & \frac{19}{3} \end{bmatrix} \quad \text{et } b^{(1)} = P^{(1)}b = \begin{bmatrix} 0 \\ 23 \\ 16 \end{bmatrix}$$

Puis à la deuxième étape :

$$P^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{3}{13} & 0 \\ -0 & -\frac{2}{13} & 1 \end{bmatrix}, \quad \text{d'où } A^{(2)} = P^{(2)}A^{(1)} = \begin{bmatrix} 1 & -\frac{1}{3} & \frac{1}{3} \\ 0 & 1 & \frac{2}{13} \\ 0 & 0 & \frac{81}{13} \end{bmatrix} \quad \text{et } b^{(2)} = P^{(2)}b^{(1)} = \begin{bmatrix} 0 \\ \frac{69}{13} \\ \frac{162}{13} \end{bmatrix}$$

et en fin à la dernière étape :

$$P^{(3)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{13}{81} \end{bmatrix}, \quad \text{d'où } A^{(3)} = P^{(3)}A^{(2)} = \begin{bmatrix} 1 & -\frac{1}{3} & \frac{1}{3} \\ 0 & 1 & \frac{2}{13} \\ 0 & 0 & 1 \end{bmatrix} \quad \text{et } x = b^{(3)} = P^{(3)}b^{(2)} = \begin{bmatrix} 0 \\ \frac{69}{13} \\ 2 \end{bmatrix}$$

Il ne reste plus qu'à résoudre le système triangulaire : $A^{(3)}x = b^{(3)}$

$$\text{ce qui donne } \begin{cases} x_3 = 2 \\ x_2 = 5 \\ x_1 = 1 \end{cases} \quad \text{et donc : } x = \begin{bmatrix} 1 \\ 5 \\ 2 \end{bmatrix}$$

Décomposons à présent A en LU . U n'est autre que la matrice $A^{(3)}$ que nous venons de déterminer : $U = A^{(3)}$.

D'autre part,

$$L = (P^{(3)}P^{(2)}P^{(1)})^{-1} = \begin{bmatrix} 6 & 0 & 0 \\ -2 & \frac{13}{3} & 0 \\ 2 & \frac{2}{3} & \frac{81}{13} \end{bmatrix}.$$

où

$$(P^{(1)})^{-1} = \begin{bmatrix} 6 & 0 & 0 \\ -2 & 1 & 0 \\ 2 & 0 & 0 \end{bmatrix}, \quad (P^{(2)})^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{13}{3} & 0 \\ 0 & \frac{2}{3} & 1 \end{bmatrix} \quad \text{et} \quad (P^{(3)})^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{81}{13} \end{bmatrix},$$

4.4.6 Programme en Matlab de la méthode de Factorisation LU

```

        clc
        clear all
        close all
        a = input('entrer la matrice a =')
        b = input('donner le second membre du système')
        n = length(b)
        aa = [a b]
        for k = 1 : n
            for i = k + 1 : n
                m(i, k) = aa(i, k)/aa(k, k);
                for j = 1 : n + 1
                    aa(i, j) = aa(i, j) - m(i, k) * aa(k, j);
                end
            end
        end
        a = aa(:, 1 : n);
        b = aa(:, n + 1);
        x(n) = b(n)/a(n, n);
        for i = n - 1 : -1 : 1
            s = 0;
            for k = i + 1 : n
                s = s + a(i, k) * x(k);
            end
            x(i) = (b(i) - s)/a(i, i);
        end
        x
    
```

4.4.7 Factorisation de Cholesky

Méthode de factorisation pour une matrice A définie positive et symétrique. Il existe une unique matrice L triangulaire inférieure dont les termes diagonaux sont strictement positifs

telle que $A = LL^T$. Les éléments de L sont, pour $i = 1$ à n

$$\begin{cases} L_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2} \\ L_{ji} = \frac{1}{a_{ii}} \left(a_{ij} - \sum_{k=1}^{i-1} l_{ik} l_{jk} \right) \quad j = i + 1 \text{ à } n \end{cases} \quad (4.4)$$

Le système à résoudre $Ax = b$ se ramène alors à la résolution de deux systèmes triangulaires :

$$\begin{cases} Ly = b \\ L^T x = y \end{cases} \quad (4.5)$$

4.4.8 Exercice d'application de la méthode de Factorisation de Cholesky

Exercice :

On considère la matrice A et le vecteur b

$$A = \begin{bmatrix} 6 & -2 & 2 \\ -2 & 5 & 0 \\ 2 & 0 & 7 \end{bmatrix}, b = \begin{bmatrix} 0 \\ 23 \\ 16 \end{bmatrix}$$

Montrer que A est symétrique et définie positive. Résoudre alors le système $Ax = b$ par la méthode de Cholesky.

Solution :

Une matrice symétrique est définie positive si ${}^t x A x \geq 0$ quel que soit le vecteur x . considérons donc un vecteur $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$. On a

$$(x_1, x_2, x_3) \begin{bmatrix} 6 & -2 & 2 \\ -2 & 5 & 0 \\ 2 & 0 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 2(x_1 - x_2)^2 + (x_1 + x_3)^2 + 2x_1^2 + 3x_2^2 + 5x_3^2 \geq 0$$

la matrice A est donc bien définie positive.

La méthode de cholesky consiste à décomposer A en $L^t L$ où L est triangulaire inférieure. On a

$$L = \begin{bmatrix} \sqrt{6} & 0 & 0 \\ -2/\sqrt{6} & \sqrt{39}/3 & 0 \\ 2/\sqrt{6} & 2\sqrt{39}/39 & 9\sqrt{13}/13 \end{bmatrix}$$

Posons ${}^t L x = y$ et résolvons le système $Ly = b$, ce qui donne :

$$\begin{cases} y_1 = 0 \\ y_2 = 23 \frac{\sqrt{39}}{13} \\ y_3 = 18 \frac{\sqrt{13}}{13} \end{cases}$$

Il ne reste plus qu'à résoudre ${}^t L x = y$, ce qui donne : $x_3 = 2$, $x_2 = 5$, $x_1 = 1$.

4.4.9 Programme en Matlab de la méthode de Factorisation de Cholesky

```

        clc; clear all; close all;
        a = input('entrer la matrice a =')
        b = input('donner le second membre du système')
        [n,m] = size(a);
        if n~ = m
            error('seulement les systèmes carrées')
        end
        for k = 1 : n - 1
        if a(k,k) < = 0
            error('pivot nul ou négatif')
        end
        a(k,k) = sqrt(a(k,k));
        a(k+1 : n,k) = a(k+1 : n,k)/a(k,k);
        for j = k+1 : n
            a(j : n,j) = a(j : n,j) - a(j : n,k) * a(j,k);
        end
        end
        a(n,n) = sqrt(a(n,n)); a = tril(a);
        y(1) = b(1)/a(1,1);
        for j = 2 : n; s = 0;
        for k = 1 : j - 1; s = s + a(j,k) * x(k); end;
        y(j) = (b(j) - s)/a(j,j); end
        a = a'; x(n) = b(n)/a(n,n);
        for i = n - 1 : -1 : 1
            s = 0;
        for k = i + 1 : n
            s = s + a(i,k) * x(k);
        end
        x(i) = (b(i) - s)/a(i,i)
        end

```

4.4.10 Méthode de Gauss-Jordin

La méthode de Gauss-Jordan est une variante de la méthode de Gauss. Elle consiste à diagonaliser la matrice A , au lieu de la triangulariser, en faisant apparaître l'unité la diagonale. Pour cela, on définit comme dans la méthode de triangularisation de Gauss, trois

suites de matrice $A^{(k)}$, $P^{(k)}$ et $b^{(k)}$, telles que :

$$\begin{aligned} A^{(1)} &= P^{(1)}A; \quad b^{(1)} = P^{(1)}b \\ &\vdots \\ A^{(k)} &= P^{(k)}A^{(k-1)}; \quad b^{(k)} = P^{(k)}b^{(k-1)} \\ &\vdots \\ A^{(n)} &= P^{(n)}A^{(n-1)}; \quad b^{(n)} = P^{(n)}b^{(n-1)} \end{aligned}$$

avec

$$P^{(1)} = \begin{bmatrix} 1/a_{11} & 0 & 0 & \cdots & 0 \\ -a_{21}/a_{11} & 1 & 0 & \cdots & 0 \\ -a_{31}/a_{11} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_{n1}/a_{11} & 0 & 0 & \cdots & 1 \end{bmatrix}; \quad A^{(1)} = \begin{bmatrix} 1 & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n-1}^{(1)} & a_{1n}^{(1)} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n-1}^{(1)} & a_{2n}^{(1)} \\ 0 & a_{32}^{(1)} & a_{33}^{(1)} & \cdots & a_{3n-1}^{(1)} & a_{3n}^{(1)} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & a_{n2}^{(1)} & a_{n3}^{(1)} & \cdots & a_{nn-1}^{(1)} & a_{nn}^{(1)} \end{bmatrix}$$

et nous pouvons écrire dans le cas générale :

$$P^{(k+1)} = \begin{bmatrix} 1 & \cdots & 0 & -a_{1k+1}^{(k)}/a_{k+1k+1}^{(k)} & \cdots & 0 & 0 \\ 0 & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 & -a_{kk+1}^{(k)}/a_{k+1k+1}^{(k)} & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 1/a_{k+1k+1}^{(k)} & \cdots & 0 & 0 \\ 0 & \cdots & 0 & -a_{k+2k+1}^{(k)}/a_{k+1k+1}^{(k)} & \ddots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & 1 & \vdots \\ 0 & \cdots & 0 & -a_{nk+1}^{(k)}/a_{k+1k+1}^{(k)} & \cdots & 0 & 1 \end{bmatrix}$$

et ainsi se suite jusqu'à $k = n - 1$, la matrice $A^{(n)} = P^{(n)}P^{(n-1)} \dots P^{(1)}A$ de sorte que $A^{(n)} = I$, alors $A^{-1} = P^{(n)}P^{(n-1)} \dots P^{(1)}$.

La résolution du système $Ax = b$ est la solution du système $A^{(n)}x = b^{(n)}$ avec $A^{(n)} = I$, a partir de ça la solution $x = b^{(n)}$.

Remarques :

La différence entre les deux méthodes Gauss et Gauss-Jordan est :

1- La matrice de passage $P^{(k)}$ est une triangulaire inférieure par rapport à la méthode de Gauss et par contre dans la méthode de Gauss-Jordan est une matrice quelconque.

2- Par la méthode de Gauss-Jordan, nous pouvons calculer l'inverse d'une matrice si elle existe.

3- Par la méthode de Gauss, nous pouvons déterminer deux matrices L et U telle que : $A = LU$ où L est une matrice triangulaire inférieure et U est une matrice triangulaire supérieure (Factorisation LU).

4.4.11 Exercice d'application de la méthode de Gauss-Jordan

Exercice :

On considère la matrice A et le vecteur b :

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 4 \end{bmatrix}, \quad b = \begin{bmatrix} 6 \\ 23 \\ 23 \end{bmatrix}$$

Résoudre le système $Ax = b$ par la méthode de Gauss-Jordan.

Solution :

Nous utilisons l'algorithme de la méthode de Gauss-Jordan :

on a à la première étape :

$$P^{(1)} = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ -\frac{1}{2} & 0 & 1 \end{bmatrix}, \quad \text{d'où } A^{(1)} = P^{(1)}A = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{3}{2} & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{7}{2} \end{bmatrix} \quad \text{et } b^{(1)} = P^{(1)}b = \begin{bmatrix} 3 \\ 20 \\ 20 \end{bmatrix}$$

Puis à la deuxième étape :

$$P^{(2)} = \begin{bmatrix} 1 & -\frac{1}{3} & 0 \\ 0 & \frac{2}{3} & 0 \\ -0 & -\frac{1}{3} & 1 \end{bmatrix}, \quad \text{d'où } A^{(2)} = P^{(2)}A^{(1)} = \begin{bmatrix} 1 & 0 & \frac{1}{3} \\ 0 & 1 & \frac{1}{3} \\ 0 & 0 & \frac{10}{3} \end{bmatrix} \quad \text{et } b^{(2)} = P^{(2)}b^{(1)} = \begin{bmatrix} -\frac{11}{3} \\ \frac{40}{3} \\ \frac{40}{3} \end{bmatrix}$$

et en fin à la dernière étape :

$$P^{(3)} = \begin{bmatrix} 1 & 0 & -\frac{1}{10} \\ 0 & 1 & -\frac{1}{10} \\ 0 & 0 & \frac{3}{10} \end{bmatrix}, \quad \text{d'où } A^{(3)} = P^{(3)}A^{(2)} = I \quad \text{et } x = b^{(3)} = P^{(3)}b^{(2)} = \begin{bmatrix} 5 \\ -12 \\ 4 \end{bmatrix}$$

La matrice inverse de la matrice A est donnée par :

$$A^{-1} = P^{(3)}P^{(2)}P^{(1)} = \frac{1}{10} \begin{bmatrix} 7 & -3 & -1 \\ -3 & 7 & -1 \\ -1 & -1 & 3 \end{bmatrix}.$$

4.4.12 Programme en Matlab de la méthode de Gauss-Jordan

```

        clc
        clear all
        close all
        a = input('entrer la matrice a =')
        b = input('donner le second membre du système')
        aa = [a b]
        [m,n] = size(aa)
        for j = 1 : m - 1
            for k = 2 : m
                if aa(j,j) == 0
                    t = aa(1,:); aa(1,:) = aa(k,:);
                    aa(k, : ) = t;
                end
            end
            for i = j + 1 : m
                aa(i, : ) = aa(i,:) - aa(j,:) * (aa(i,j)/aa(j,j));
            end
        end
        for j = m : -1 : 2
            for i = j - 1 : -1 : 1
                aa(i, : ) = aa(i,:) - aa(j,:) * (aa(i,j)/aa(j,j));
            end
        end
        for l = 1 : m
            aa(l, : ) = aa(l,:)/aa(l,l);
            x(l) = aa(l,n);
        end
        aa
        x'

```

4.5 Les méthodes itératives

Les méthodes itératives s'inspirent de l'analyse. Ce sont des méthodes de point fixe. Elles cherchent à définir une suite qui verge vers la solution. Ici, au bout d'un nombre fini d'étapes, on n'obtient jamais, par nature, la solution exacte. Pourtant certaines méthodes itératives s'en approchent très vite. Ces méthodes sont les seules à pouvoir fournir des résultats précis pour les très gros systèmes.

Nous cherchons à résoudre le système linéaire (4.2), que l'on peut noter matriciellement : $AX = b$

Pour résoudre ce système il faut déterminant de la matrice A est différent de zéro.

4.5.1 Principe des méthodes itératives

L'objectif des méthodes itératives est de construire, à partir d'un vecteur $X^{(0)}$, une suite de vecteurs $X^{(1)}, X^{(2)}, X^{(3)}, \dots, X^{(n)}, \dots$ qui converge vers la solution exacte du système (4.2).

Ayant choisi une norme $\|\cdot\|$ sur K^n , cette dernière condition retraduit par :

$$\lim_{n \rightarrow \infty} \|X^{(n)} - X\| = 0 \quad (4.6)$$

Le critère (4.6) a l'inconvénient d'utiliser l'inconnue x . Désignons toujours par $\|\cdot\|$ la norme subordonnée associée.

$$X^{(n)} - X = A^{-1} (AX^{(n)} - AX)$$

alors

$$\|X^{(n)} - X\| \leq \|A^{-1}\| \|AX^{(n)} - b\|$$

De même il est clair que

$$\|AX^{(n)} - b\| \leq \|A\| \|X^{(n)} - X\|$$

Par suite (4.6) est équivalent à :

$$\lim_{n \rightarrow \infty} \|AX^{(n)} - b\| \quad (4.7)$$

Le critère (4.7) ne dépend que des données du système (4.2).

Définition : dans ce qui suit nous allons étudier les processus itératifs linéaires du type suivant :

$$\begin{cases} X^{(k+1)} = M X^{(k)} + N \\ X^{(0)} \text{ arbitraire} \\ M \in M_n(K), N \in K^n \end{cases} \quad (4.8)$$

Nous allons chercher les conditions sur la matrice M et le vecteur N pour que le processus itératifs soit convergent et que la limite soit une solution du système (4.2). Si la suite $X^{(k)}$ converge vers X , alors nous pouvons écrire $X = M X + N$ ou $(I - M) X = N$

ceci nous amène à ne chercher que des méthodes itératives pour les quelles $I - M$ est inversible.

4.5.2 Méthode de Jacobi

On décompose A sous la forme : $A = D - E - F$ où quels que soient les indices i et j on a :

$D(i, j) = \delta_{ij} A(i, j)$; $E(i, j) = -A(i, j)$ si $j < i$ et $E(i, j) = 0$ si non; $F(i, j) = -A(i, j)$ si $j > i$ et $F(i, j) = 0$ sinon.

Pour développer cette méthode il faut que D est inversible. par compariasant avec le système (4.8), on obtient :

$$AX = b \text{ équivariant } (D - E - F)X = b$$

$$\text{équivariant aussi } DX = (E - F)X + b$$

alors

$$X = D^{-1}(E + F)X + D^{-1}b \quad (4.9)$$

donc

$$M = D^{-1}(E + F)$$

$$N = D^{-1}b$$

M est appelée la matrice de Jacobi et sera noté J .

4.5.3 Exercices d'application de la méthode de Jacobi

Exercice 01 :

Trouver la matrice de Jacobi J du système $AX = b$ où $A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 2 \end{bmatrix}$ et $b = \begin{bmatrix} 3 \\ 5 \\ 1 \end{bmatrix}$

Solution 01 :

La matrice $D = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 2 \end{bmatrix}$ est inversible où $D^{-1} = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{4} & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix}$, la matrice $E = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$ et $F = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}$.

alors $E + F = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ donc la matrice de Jacobi s'écrit :

$$J = D^{-1}(E + F) = \begin{bmatrix} 0 & -\frac{1}{2} & 0 \\ -\frac{1}{4} & 0 & -\frac{1}{4} \\ 0 & -\frac{1}{2} & 0 \end{bmatrix}$$

Nous pouvons écrire la solution du système (4.2) par la méthode de Jacobi comme suit :

$$x_i^{(k+1)} = -\sum_{j \neq i}^n \frac{a_{ij}}{a_{ii}} x_j^{(k)} + \frac{b_i}{a_{ii}}, \quad \forall i = 1, 2, \dots, n \quad (4.10)$$

ainsi $x_1^{(k+1)}$ est calculé à partir de $x_2^{(k)}, x_3^{(k)}, \dots, x_n^{(k)}$, en annulant la 1^{er} composante; $x_2^{(k+1)}$ est calculé à partir de $x_1^{(k)}, x_3^{(k)}, \dots, x_n^{(k)}$, en annulant la deuxième composante; et ainsi de suit.

Les n composantes de $x^{(k)}$ doivent être conservées en mémoire jusqu'au calcul de $x_n^{(k+1)}$.

Exercice 02 :

Résoudre le système $Ax = b$ par la méthode de Jacobi où $A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 2 \end{bmatrix}$, $b = \begin{bmatrix} 3 \\ 5 \\ 1 \end{bmatrix}$

et le vecteur initial $x^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

Solution 02 :

L'algorithme de la méthode de Jacobi s'écrit comme : $x^{(k+1)} = J x^{(k)} + D^{-1}b$

La matrice de Jacobi : $J = D^{-1}(E + F) = \begin{bmatrix} 0 & -\frac{1}{2} & 0 \\ -\frac{1}{4} & 0 & -\frac{1}{4} \\ 0 & -\frac{1}{2} & 0 \end{bmatrix}$ et $D^{-1}b = \begin{bmatrix} \frac{3}{2} \\ \frac{5}{4} \\ \frac{1}{2} \end{bmatrix}$

On donne les résultats sur le tableau suivant :

k	0	1	2	3	4	5	6	7	8
$x_1^{(k)}$	0	1.5	0.875	1.125					
$x_2^{(k)}$	0	1.25	1.0625	0.9375					
$x_3^{(k)}$	0	-0.125	0.125	-0.03125					

alors la solution est $x^T = (1, 1, 0)$.

4.5.4 Programme en Matlab de la méthode de Jacobi

```

clc
clear all
close all

a = input('entrer la matrice a =')
b = input('donner le second membre du système')
n = length(b)
x0 = zeros(n,1);
err = 1;
while err > 10^-6
    for j = 1:n
        x(j) = (b(j) - a(j, [1:j-1, j+1:n]) * x0([1:j-1, j+1:n]))/a(j,j);
    end
    err = max(abs(x' - x0));
    x0 = x'
end
r = x'

```

4.5.5 Méthode de Gauss- Seidel

A étant toujours décomposée sous la forme $A = D - E - F$ comme pour la méthode de Jacobi, avec toujours D inversible, on remarque que $D - E$ est aussi inversible (puisque $\det(D - E) = \det(D) \neq 0$).

$AX = b$ équivalent $(D - E - F)X = b$

équivalent aussi $(D - E)X = FX + b$

alors

$$X = (D - E)^{-1}FX + (D - E)^{-1}b \quad (4.11)$$

donc

$$\begin{aligned} M &= (D - E)^{-1}F \\ N &= (D - E)^{-1}b \end{aligned}$$

dans cette méthode, la matrice M est appelée la matrice de Gauss-Seidel et sera noté G .

4.5.6 Exercices d'application de la méthode de Gauss- Seidel

Exercice 01 :

Trouver la matrice Gauss-Seidel du système $AX = b$ où $A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 2 \end{bmatrix}$ et $b = \begin{bmatrix} 3 \\ 5 \\ 1 \end{bmatrix}$

Solution 01 :

La matrice $D - E = \begin{bmatrix} 2 & 0 & 0 \\ 1 & 4 & 0 \\ 0 & 1 & 2 \end{bmatrix}$ est inversible et $F = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}$.

On utilise la méthode de Gauss-Jordan pour trouver l'inverse de $(D - E)$.

la première matrice de passage $P^{(1)} = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $(D - E)^{(1)} = P^{(1)}(D - E) =$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 1 & 2 \end{bmatrix},$$

la deuxième matrice de passage $P^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{4} & 0 \\ 0 & -\frac{1}{4} & 1 \end{bmatrix}$, $(D - E)^{(2)} = P^{(2)}(D - E)^{(1)} =$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix},$$

et la troisième matrice de passage $P^{(3)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix}$, $(D - E)^{(3)} = P^{(3)}(D - E)^{(2)} =$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

alors $(D - E)^{-1} = P^{(3)}P^{(2)}P^{(1)} = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ -\frac{1}{8} & \frac{1}{4} & 0 \\ \frac{1}{16} & -\frac{1}{8} & \frac{1}{2} \end{bmatrix}$

donc la matrice de Gauss-Seidel M s'écrit :

$$G = (D - E)^{-1}F = \begin{bmatrix} 0 & -\frac{1}{2} & 0 \\ 0 & \frac{1}{8} & -\frac{1}{4} \\ 0 & -\frac{1}{16} & \frac{1}{8} \end{bmatrix}$$

Nous pouvons écrire l'expression de la solution du système (4.2) par la méthode de Gauss-Seidel comme suit :

$$x_i^{(k+1)} = -\sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{(k+1)} - \sum_{j=i+1}^n \frac{a_{ij}}{a_{ii}} x_j^{(k)} + \frac{b_i}{a_{ii}}, \quad \forall i = 1, 2, \dots, n \quad (4.12)$$

ainsi la i ième composante $x^{(k+1)}$ est calculée à partir des $(i - 1)$ premières composantes de $x^{(k+1)}$ et des $(n - i - 1)$ dernières composantes de $x^{(k)}$.

Exercice 02 :

Résoudre le système $Ax = b$ par la méthode de Gauss-Seidel où $A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 2 \end{bmatrix}$,

$$b = \begin{bmatrix} 3 \\ 5 \\ 1 \end{bmatrix} \text{ et le vecteur initial } x^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Solution 02 :

L'algorithme de la méthode de Gauss-Seidel s'écrit comme : $x^{(k+1)} = G x^{(k)} + (D - E)^{-1} b$

La matrice de Gauss-Seidel s'écrit : $G = (D - E)^{-1} F = \begin{bmatrix} 0 & -\frac{1}{2} & 0 \\ 0 & \frac{1}{8} & -\frac{1}{4} \\ 0 & -\frac{1}{16} & \frac{1}{8} \end{bmatrix}$ et $(D - E)^{-1} b =$

$$\begin{bmatrix} \frac{3}{2} \\ \frac{7}{8} \\ \frac{1}{16} \end{bmatrix}$$

On donne les résultats sur le tableau suivant :

k	0	1	2	3	4	5	6
$x_1^{(k)}$	0	1.5	1.0625	1.015625	1.00390625	1.0009765625	1.0002441406
$x_2^{(k)}$	0	0.875	0.96875	0.9921875	0.998046875	0.9995117187	0.9998779297
$x_3^{(k)}$	0	0.0625	0.015625	-0.00390625	0.0009765625	0.0002441406	0.0000610351

alors la solution est $x^T = (1, 1, 0)$.

4.5.7 Programme en Matlab de la méthode de Gauss- Seidel

```

        clc
        clear all
        close all
        a = input('entrer la matrice a =')
        b = input('donner le second membre du système')
        n = length(b)
        x0 = zeros(n,1);
        err = 1;
    while err > 10-6
        for j = 1 : n
            if j = 1
                x(1) = (b(1) - a(1, 2 : n) * x0(2 : n))/a(1,1);
            elseif j = n
                x(n) = (b(n) - a(n, 1 : n - 1) * x0(1 : n - 1))/a(n, n);
            else
                x(j) = (b(j) - a(j, 1 : j - 1) * x(1 : j - 1) - a(j, j + 1 : n) * x0(j + 1 : n))/a(j, j);
            end
        end
        err = max(abs(x' - x0));
        x0 = x';
    end
    r = x'

```

4.6 La convergence des méthodes itératives

La convergence des méthodes itératives où l'expression (4.8) est liée à la notion de rayon spectral de la matrice M (dans la méthode de Jacobi $M = J$ et dans la méthode de Gauss Seidel $M = G$). Nous étudions, maintenant, cette notion.

4.6.1 Rayon Spectral

On appelle rayon spectral de $M \in M_n(K)$ et note $\rho(M)$ le plus grand des modules des vecteurs propres de M :

$$\rho(M) = \max \{|\lambda|; \lambda \text{ valeur propre de } M\}$$

Exemple :

Soit $M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & 2 \end{bmatrix}$ une matrice diagonale donc le rayon spectral de cette matrice

est $\rho(M) = \max_{1 \leq i \leq 3} \{|\lambda_i|\}$

$$\rho(M) = \max\{1, 2, 3\} = 3$$

Nous utilisons le théorème suivant pour étudier la convergence des méthodes itératives.

Théorème :

Soit $M \in M_n(K)$, $(I - M)$ inversible et $N \in K^n$. La méthode itérative :

$$\begin{cases} X^{(k+1)} = MX^{(k)} + N \\ X^{(0)} \in K^n \end{cases}$$

converge, quelque soit le vecteur initial $X^{(0)}$, si et seulement si $\rho(M) < 1$.

Remarques :

1- En pratique, le calcul de $\rho(M)$ est très compliqué, il suffit alors de vérifier si $\|M\| < 1$ puisque $\rho(M) \leq \|M\|$.

2- Les méthodes de Jacobi et Gauss-Seidel converge si l'une des conditions suivantes est vérifiée :

a- $\|M\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |M_{ij}| < 1$

b- $\|M\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n |M_{ij}| < 1$

c- $\|M\| = \sqrt{\sum_{i=1}^n \sum_{j=1}^n |M_{ij}|^2} < 1$

3- Si $|M_{ij}| < \frac{1}{n}$; $\forall i, j$ où n est la dimension du système, alors les algorithmes de Jacobi et Gauss-Seidel convergent.

4- Si $|a_{ii}| > \sum_{j \neq i}^n |M_{ij}| \quad \forall i$

où $|a_{ii}| > \sum_{i \neq j}^n |M_{ij}| \quad \forall j$ alors le processus converge.

(Dans ce cas on dit que la matrice A est à diagonale dominante).

5- La méthode de Gauss-Seidel présente l'avantage suivant par rapport à celle de Jacobi. On n'est pas obligé de connaître toutes les composantes de $X^{(k)}$ pour pouvoir calculer $X^{(k+1)}$, et aussi la convergence de la méthode de Gauss-Seidel plus rapide que la méthode de Jacobi.

4.7 Série des exercices

Exercice 4 - 1 :

Résoudre par la méthode de Gauss les systèmes linéaires suivants :

$$1- \begin{cases} 2x_1 + 3x_2 - x_3 = 5 \\ 4x_1 + 4x_2 - 3x_3 = 3 \\ -2x_1 + 3x_2 - x_3 = 1 \end{cases}, \quad 2- \begin{cases} x_1 + x_2 + x_3 = 1 \\ 2x_2 + 4x_3 = 1 \\ 3x_1 + 3x_2 + x_3 = 1 \end{cases}, \quad 3- \begin{cases} 2x_1 + x_2 + 4x_4 = 2 \\ -4x_1 - 2x_2 + 3x_3 - 7x_4 = -9 \\ 4x_1 + x_2 - 2x_3 + 8x_4 = 2 \\ -3x_2 - 12x_3 - x_4 = 2 \end{cases}$$

Exercice 4 - 2 :

On considère les matrices :

$$A = \begin{bmatrix} 2 & 1 & 3 \\ 3 & 2 & 5 \\ 3 & 7 & 3 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 2 & 3 \\ 3 & 2 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix}$$

- 1)- Trouver la décomposition LU de la matrice A par la méthode de Gauss.
- 2)- Trouver A^{-1} et B^{-1} par la méthode de Gauss –Jordan.
- 3)- En déduire $\det(A)$ et $\det(B)$.
- 4)- En déduire les solutions de $AX = (4, 6, -1)^t$ et $BX = (-2, 0, 0, 2)^t$.

Exercice 4 - 3 :

On souhaite résoudre le système linéaire suivant $AX = b$, par la méthode de Gauss. La matrice A et le vecteur b sont définie par

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 5 \\ 4 & 6 & 8 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix} \quad (4.13)$$

- 1) Que se passe t-il si on applique l'algorithme de Gauss ?
- 2) Pour résoudre ce problème, on introduit la matrice de permutation suivante

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Ecrire le système équivalent au système précédent qui a PA comme matrice associée.

- 3) Appliquer la méthode de Gauss à ce nouveau système et calculer la factorisation LU associée.
- 4) Résoudre le système à partir de la factorisation LU obtenue.

Exercice 4 - 4 :

On considère les matrices :

$$A = \begin{bmatrix} -14 & 18 & -27 \\ 4 & -4 & 7 \\ 12 & -12 & 22 \end{bmatrix}, \quad B = \begin{bmatrix} -56 & -14 & 1 \\ 12 & 4 & 0 \\ 48 & 12 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 296 \\ -64 \\ -240 \end{bmatrix}$$

- I)- 1 – Décomposer B en LU où L et U sont deux matrices triangulaires inférieure et supérieure respectivement.
- I)- 2 – Inverser la matrice B par la méthode de Gauss-Jordan.
- I)- 3 – En déduire la solution du système $Bx = b$.
- I)- 4 – On considère la suite de vecteurs : $Y^{(n)} = A^n Y^{(0)}$ et $Y^{(0)} = (1, 0, 0)^T$
 - I)- 4 - 1 – Calculer $Y^{(1)}, Y^{(2)}$ et $Y^{(3)}$.
 - I)- 4 - 2 – Donner le polynôme caractéristique de A qui définir par la relation suivante : $P_3(\lambda) = \sum_{i=0}^3 x_i \lambda^i$ où $x_0 = 1$ et $x_i, i = 1, \dots, 3$ sont des solutions du système qui s'écrit comme $(Y^{(2)}, Y^{(1)}, Y^{(0)}) (x_1, x_2, x_3)^T = -Y^{(3)}$.

Exercice 4 - 5 :

1)- Montrer que la matrice A est symétrique et définie positive.

$$A = \begin{bmatrix} 6 & -2 & 2 \\ -2 & 5 & 0 \\ 2 & 0 & 7 \end{bmatrix}, \quad (4.14)$$

2)- Résoudre alors le système $AX = b$ par la méthode de Choleski où $b^t = (0, 23, 16)$.

Exercice 4 - 6 :

On considère la matrice A et le vecteur b :

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 4 \end{bmatrix}, \quad b = \begin{bmatrix} 8 \\ 9 \\ 19 \end{bmatrix} \quad (4.15)$$

01)- Résoudre le système par la méthode de Jacobi et par la méthode de Gauss-Seidel. On partira du vecteur $X^{(0)} = (0, 0, 0)^t$.

02)- Retrouver la solution précédente en inversant A par la méthode de Gauss-Jordan.

Exercice 4 - 7 :

On considère la matrice A et le vecteur b :

$$A = \begin{bmatrix} -6 & 18 & -27 \\ 4 & -4 & 7 \\ 12 & -12 & 22 \end{bmatrix}, \quad b = \begin{bmatrix} 296 \\ -64 \\ -240 \end{bmatrix} \quad (4.16)$$

01)- Montrer, sans le calcul des valeurs propres, que la méthode de Jacobi correspondante à (4.16) converge quelque soit $X^{(0)}$.

02)- Résoudre le système $AX = b$ par la méthode de triangularisation de Gauss. Donner la décomposition LU de A où les matrices L et U sont triangulaires inférieure et supérieure respectivement. En déduire A^{-1} .

03)- Résoudre le même système, avec une précision $\varepsilon = 10^{-2}$, par la méthode de Jacobi. On partira de $X^{(0)} = (1, 0, 0)^t$.

04)- Résoudre le même système par la méthode de Gauss-Seidel. On prendra le même vecteur initial qu'à la question précédente et avec la même précision.

Exercice 4 - 8 :

On considère le système linéaire :

$$Ax = b \quad (4.17)$$

où $A = (a_{ij})$ est une matrice réelle, carrée, d'ordre n , inversible et d'éléments diagonaux a_{ii} , $i = 1, \dots, n$ non nuls, x et b donnée sont deux vecteurs de R^n notés : $x^t = (x_1, x_2, \dots, x_n)$ et $b^t = (b_1, b_2, \dots, b_n)$.

On décompose A en $A = M - N$ où M et N sont les matrices d'éléments :

$$M_{ij} = \begin{cases} a_{ij} & i \geq j \\ 0 & i < j \end{cases}, \quad N_{ij} = \begin{cases} -a_{ij} & i > j \\ 0 & i \geq j \end{cases},$$

1- Vérifier que le système (4.17) est équivalent au système :

$$x = M^{-1}Nx + M^{-1}b \quad (4.18)$$

On associe à ce système le schéma itératif :

$$x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b \quad (4.19)$$

Ecrire une condition suffisante de convergence de (4.19).

2- Etablir une relation de récurrence permettant de calculer les éléments g_{ij} et x_j des matrices $G = M^{-1}N$ et $x = M^{-1}b$ en fonction des éléments a_{ij} de A et b_i de b .

3- Montrer que :

$$a_{ii}x_i^{(k+1)} = - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k+1)} + b_i, \quad i = 1, 2, \dots, n \quad (4.20)$$

4- Application : on considère le système $Ax = b$ avec :

$$A = \begin{bmatrix} 6 & -2 & 2 \\ -2 & 5 & 0 \\ 2 & 0 & 7 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 23 \\ 16 \end{bmatrix}, \quad x^{(0)} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

En partant du vecteur initial $x^{(0)t} = (1, 0, 0)$, déterminer la solution de ce système à l'aide du schéma (4.19).

Chapitre 5

Les méthodes numériques à un pas

5.1 Définition les Méthodes à un pas

Les méthodes à un pas sont des méthodes de résolution numérique qui peuvent s'écrire sous la forme suivante :

$$y_{n+1} = y_n + h_n \Phi(t_n, y_n, h_n), \quad \forall 0 \leq n < N \quad (5.1)$$

où $\Phi : [t_0, t_0 + T] \times R \times R \rightarrow R$ est une fonction qu'on supposera continue. Dans la pratique la fonction peut n'être définie que sur une partie de la forme $[t_0, t_0 + T] \times J \times [0, \delta]$ où J est un intervalle de R (de sorte en particulier que $[t_0, t_0 + T] \times J$ soit contenu dans le domaine de définition de l'équation différentiel). Dans toutes les méthodes numériques développées par la suite, on subdivise l'intervalle $[t_0, t_0 + T]$ en N intervalle de longueur $h = \frac{(t_0+T)-t_0}{N} = \frac{T}{N}$, telle que $h = \max_n h_n$, limités par les points $t_n = t_0 + nh$, $0 \leq n < N$.

Les types de l'erreur

Pour l'erreur on distingue deux cas :

L'erreur locale : $e_n = y(t_n) - y_n$.

L'erreur globale : $\varepsilon(h) = \max_{0 \leq n \leq N} |e_n|$.

5.2 La méthode d'Euler

Définition de la méthode d'Euler

La méthode d'Euler (ou méthode de la tangente) est une méthode simple de résolution d'une équation différentielle ordinaire (EDO). Comme son nom l'indique, elle est due au mathématicien et physicien suisse Euler (1707-1783). On considère le problème de Cauchy suivant :

$$\begin{cases} y' = f(t, y(t)) \\ y(t_0) = u_0 \end{cases} \quad (5.2)$$

où f une fonction définie sur une partie de R^2 où $u_0 \in R$ on subdivise l'intervalle $[t_0, t_0 + T]$ par des points t_0, t_1, \dots, t_N telle que : $t_0 < t_1 < \dots < t_n < t_{n+1} < \dots < t_N = t_0 + T$ est une suite nombre réelle équiartis tel que :

$$h_n = t_{n+1} - t_n$$

avec le pas $h = \max_n h_n$

$$t_n = t_0 + nh, \text{ pour tout } 0 \leq n \leq N$$

La solution du problème de Cauchy vérifié :

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(t, y(t)) dt$$

Notation :

★ y_n une approximation de $y(t_n)$.

★ y_{n+1} une approximation de $y(t_{n+1})$.

La formule générale d'euler

- La méthode d'Euler consiste à calculer, par récurrence des valeurs approchées de respectivement au moyen de la formule suivante :

$$y_{n+1} = y_n + hf(t_n, y_n). \quad (5.3)$$

Démonstration de La formule générale d'euler : On considère que $y(t)$ est une solution exacte de problème de Cauchy et continue sur $[t_0, t_0 + T]$

$$\text{en } t_0 : y'(t_0) = f(t_0, y(t_0))$$

$$\text{et on a } t_{n+1} = t_n + h \rightarrow t_1 = t_0 + h$$

la solution d'Euler en cette point est

$$y(t_1) = y_0 + hf(t_0, y_0) \quad (5.4)$$

pour le deuxième point $t_2 = t_1 + h$ s'écrit :

$$y(t_2) = y_1 + hf(t_1, y_1) \quad (5.5)$$

alors on peut écrire la formule suivante :

$$y(t_{n+1}) = y_n + hf(t_n, y_n) \quad (5.6)$$

L'expression de l'erreur de la méthode de Euler est

$$\varepsilon_1(t) = \frac{h^2}{2} f'(t, y)$$

et aussi s'écrit

$$\varepsilon_1(t) = \frac{h^2}{2} \left(\frac{\partial f(t_n, y_n)}{\partial t} + \frac{\partial f(t_n, y_n)}{\partial y} f(t_n, y_n) \right) \quad (5.7)$$

Justification géométrique :

Le point (t_{n+1}, y_{n+1}) est sur la droite contenant (t_n, y_n) et de pente $f(t_n, y_n)$ or, $f(t_n, y_n)$ est le pente de la tangente à la courbe de solution du problème de Cauchy suivante :

$$\begin{cases} y' = f(t, y(t)) \\ y(t_0) = u_0 \end{cases}$$

La solution approchée y s'obtient graphiquement en traçant pour chaque n les segment joignant $(t_n, y_n), (t_{n+1}, y_{n+1})$

On construit de même une solution approcher sur en prenant des pas $h_n < 0$.

Précision de la méthode d'Euler

La méthode d'Euler est une méthode du premier ordre, c'est -à-dire que l'erreur au point t_n s'exprime par l'inégalité $|y_n - y(t_n)| \leq kh$ où y_n est la valeur approchée définie par l'algorithme d'Euler, $y(t_n)$ est la valeur exacte de la solution de problème de Cauchy au point $t = t_n = t_0 + nh$ et k une constance indépendante de n et de h .

Démonstration : Soit la formule d'Euler $y_{n+1} = y_n + hf(t_n, y_n)$.

Erreur due au schéma numérique (Euler) :

-La solution exacte et le schéma numérique vérifient :

$$y(t_{n+1}) = y(t_n) + hf(t_n, y(t_n)) + \varepsilon_n$$

$$y_{n+1} = y_n + hf(t_n, y_n).$$

Alors, comme $e_n = y(t_n) - y_n$, on obtient $e_{n+1} = e_n + h(f(t_n, y(t_n)) - f(t_n, y_n)) + \varepsilon_n$.

-Si F est localement lipchitzienne en y uniformément en t (hypothèse du théorème de Cauchy-Lipchitz), on a

$$|f(t_n, y(t_n)) - f(t_n, y_n)| \leq L |e_n|$$

et

$$|e_{n+1}| \leq |e_n| (1 + Lh) + |\varepsilon_n|.$$

Deux lemmes intermédiaires

Lemme 1 Soit $(\theta_n)_{n \geq 0}$ une suite positive vérifiant

$\forall 0 \leq n \leq N, \theta_{n+1} \leq a \theta_n + \alpha$; avec $a > 0$ et $\alpha > 0$ alors, $\forall 1 \leq n \leq N + 1$,
 $\theta_n \leq a^n \theta_0 + \sum_{i=0}^{n-1} a^i \alpha = a^n \theta_0 + \alpha \frac{1-a^n}{1-a}$.

Lemme 2 De plus, si $a = 1 + \rho$ avec $\rho > 0$, comme $(1 + \rho)^n \leq e^{n\rho}$, on a

$$\theta^n \leq e^{n\rho} \theta_0 + \frac{\alpha}{\rho} (e^{n\rho} - 1); \quad 1 \leq n \leq N + 1.$$

on a $\varepsilon_n = \frac{h^2}{2} y''(\xi_n)$, on pose $M_2 = \sup_{[t_0, t_0+T]} |y''(t)|$ alors $\varepsilon_n \leq \frac{h^2}{2} M_2$.

on a, pour tout $0 \leq n \leq N - 1$

$$|e_{n+1}| \leq |e_n| + (1 + Lh) |\varepsilon_n|,$$

$$\leq |e_n| + (1 + Lh) \frac{M_2 h^2}{2}. \quad (\text{car } |\varepsilon_n| \leq \frac{M_2 h^2}{2})$$

on applique le Lemme 2 avec $\rho = Lh$ et $\alpha = \frac{M_2 h^2}{2}$.

$$|e_n| \leq e^{nLh} |e_0| + \frac{M_2 h^2}{2L} (e^{nLh} - 1), \quad 1 \leq n \leq N.$$

mais, pour $1 \leq n \leq N, nh \leq Nh = T$, et

$$|e_n| \leq e^{LT} |e_0| + \frac{M_2}{2} \frac{e^{LT} - 1}{L} h, \quad \forall 1 \leq n \leq N,$$

donc la majoration de l'erreur est vérifier

ainsi, si $e_0 = 0$:

$$\varepsilon(h) \leq \frac{M_2}{2} \frac{e^{LT} - 1}{L} h$$

alors

$$K = \frac{M_2}{2} \frac{e^{LT} - 1}{L}$$

Remarque important :

La résolution de l'équation du premier ordre avec condition initial (problème de Cauchy) est universellement connue. Mais pour résoudre l'équation différentielle du deuxième ordre $y'' = f(t, y(t))$ Euler ce faire, il réécrit cette équation sous forme d'un système de deux équations du premier ordre : $\frac{dy}{dt} = p, \frac{dp}{dt} = v(t, y, p)$.

Pour que le problème soit posé correctement il faut donner une condition initiale supplémentaire $p_{t=t_0} = v$, en plus de la condition initiale évidente $y(t_0) = u_0$ c'est-à-dire que :

$$\begin{cases} y'' = f(t, y(t)) \\ y(t_0) = u_0 \end{cases} \Rightarrow \begin{cases} y' = p, \\ p' = v(t, y, p) \\ p_{t=t_0} = y'(t_0) = v \\ y(t_0) = u_0 \end{cases}$$

Par exemple :

On a l'équation différentielle suivant :

$$ay'' + h \sin y = 0 \quad (5.8)$$

Pour résoudre cette équation on posant :

$y' = \frac{dy}{dt} = g(y)$. on a : $y'' = g'y' = g'g$ et (5.8) devient :

$ag'g + h \sin y = 0$.

qui s'écrit encore : $g' = -\frac{h}{a} \cdot \frac{\sin y}{g} = f(t, y)$.

Expression d'Euler explicite (progressif)

$$\begin{cases} y_{n+1} = y_n + hf(t_n, y_n) \\ y(t_0) = u_0 \end{cases} \quad (5.9)$$

Cette expression, elle permet de obtenir directement y_{n+1} à partir de y_n .

Expression d'Euler implicite (Rétrograde)

$$\begin{cases} y_{n+1} = y_n + hf(t_{n+1}, y_{n+1}) \\ y(t_0) = u_0 \end{cases}$$

Cette expression, elle se ramène à la résolution d'un problème non linéaire pour chaque n . Ceci nécessitera de choisir une méthode numérique pour résoudre une équation algébrique non-linéaire pour chaque n .

Remarque : On dit que les schémas d'Euler progressif et rétrograde sont des schémas à un pas puisqu'elles ne font intervenir les solutions approchées qu'aux points t_n et t_{n+1} et la méthode d'Euler explicite et implicite est du premier ordre.

5.2.1 Exercice d'application de la méthode d'Euler

Exercice :

On considère le problème de Cauchy

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

sur l'intervalle $[0; 10]$.

1. Calculer la solution exacte du problème de Cauchy.
2. Soit Δt le pas temporel. Écrire la méthode d'Euler explicite pour cette équation différentielle ordinaire (EDO).
3. En déduire une forme du type

$$y_{k+1} = g(t; k)$$

avec $g(t; k)$ à préciser (autrement dit, l'itérée en t_k ne dépend que de t et k et ne dépend pas de y_k).

4. Utiliser la formulation ainsi obtenue pour trouver les solutions sur l'intervalle $[0, 10]$ avec la méthode d'Euler où $\Delta t = 2.5, 0.5$.

Solution :

1. Il s'agit d'une EDO à variables séparables. L'unique solution constante est $y(t) = 0$, toutes les autres solutions sont du type $y(t) = Ce^{-t}$. Donc l'unique solution du problème de Cauchy est $y(t) = e^{-t}$ définie pour tout $t \in \mathfrak{R}$.

2. La méthode d'Euler est une méthode d'intégration numérique d'EDO du premier ordre de la forme $y'(t) = F(t; y(t))$ C'est une méthode itérative : la valeur y à l'instant $t + \Delta t$ de déduisant de la valeur de y à l'instant t par l'approximation linéaire

$$y(t + \Delta t) = y(t) + y'(t)\Delta t = y(t) + F(t; y(t))\Delta t$$

En choisissant un pas de discrétisation Δt , nous obtenons une suite de valeurs $(t_k; y_k)$ qui peuvent être une excellente approximation de la fonction $y(t)$ avec

$$\begin{cases} t_{k+1} = t_1 + k\Delta t \\ y_{k+1} = y_k + F(t_k, y_k)\Delta t \end{cases}$$

La méthode d'Euler explicite pour cette EDO s'écrit donc

$$y_{k+1} = y_k - y_k\Delta t$$

3. En procédant par récurrence sur k , on obtient

$$y_k + 1 = (1 - \Delta t)^{k+1}$$

4. On a donc

– si $\Delta t = 2.5$ alors

$$y_k = \left(\frac{-3}{2}\right)^{k+1}$$

– si $\Delta t = 0.5$ alors

$$y_k = \left(\frac{1}{2}\right)^{k+1}$$

5.2.2 Programme en Matlab de la méthode d'Euler

```

    clc
    clear all
    close all
    a = input('la valeur minimale de l\'intervalle')
    b = input('la valeur maximale de l\'intervalle')
    n = input('le nombre des intervalles')
    y(1) = input('la valeur initiale de la solution')
    h = (b - a)/n
    f = -z;
    for i = 0 : n
        x(i + 1) = a + i * h
    end
    for i = 1 : n
        f1 = subs(f, t, x(i), z, y(i))
        y(i + 1) = y(i) + f1 * h
    end
    x
    y

```

5.3 Méthode d'Euler modifier

la méthode d'Euler modifier est une méthode qui est obtenue d'après le développement de la méthode supérieur à celui de la méthode d'Euler donc on d'après la formule de Taylor :

$$y_{n+1} = y_n + hy'_n + \frac{h^2}{2}y''_n + \varepsilon_n ; 0 \leq n \leq N$$

où

$$y''_n = \frac{y'_{n+1} - y'_n}{h}$$

alors

$$y_{n+1} = y_n + hy'_n + \frac{h^2}{2} \left(\frac{y'_{n+1} - y'_n}{h} \right) + \varepsilon_n$$

$$= y_n + hy'_n + \frac{h}{2}(y'_{n+1} - y'_n) + \varepsilon_n$$

$$y_{n+1} = y_n + \frac{h}{2}(y'_{n+1} + y'_n) + \varepsilon_n$$

et on peut écrire

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

donc la forme générale d'Euler modifier :

$$\begin{cases} y_{n+1} = y_n + hf(t_n, y_n) \\ y_{n+1} = y_n + \frac{h}{2}(f(t_{n+1}, y_{n+1}) + f(t_n, y_n)) \end{cases} \quad (5.10)$$

5.3.1 Exercice d'application de la méthode d'Euler modifier

Exercice :

Faire trois itérations avec $h = 0.1$ par la méthode de Runge-Kutta d'ordre 4 pour l'équation différentielle suivante :

$$y'(t) = t^2 + y^2(t) + 1 \quad (y(1) = 0)$$

Solution :

La forme générale d'Euler modifier

$$\begin{cases} y_{n+1} = y_n + hf(t_n, y_n) \\ y_{n+1} = y_n + \frac{h}{2}(f(t_{n+1}, y_{n+1}) + f(t_n, y_n)) \end{cases}$$

On a que $h = 0.1$, $t_0 = 1$, $y_0 = 0$ et que $f(t_n, y_n) = t_n^2 + y_n^2 + 1$.
donc $t_1 = t_0 + h = 1.1$, $t_2 = t_1 + h = 1.2$, $t_3 = t_2 + h = 1.3$.

- Première itération

$$\hat{y} = y_0 + hf(t_0, y_0) = 0 + 0.1 \times f(1, 0) = 0, 2$$

$$y_1 = y_0 + h/2(f(t_0, y_0) + f(t_0 + h, \hat{y})) = 0 + 0.05 \times (f(1, 0) + f(1.1, 0.2)) = 0.2125 \text{ à } t_1 = 1.1$$

- Deuxième itération

$$\hat{y} = y_1 + hf(t_1, y_1) = 0.2125 + 0.1 \times f(1.1, 0.2125) = 0.4380156$$

$$y_2 = y_1 + h/2(f(t_1, y_1) + f(t_1 + h, \hat{y}))$$

$$y_2 = 0.2125 + 0.05 \times (f(1.1, 0.2125) + f(1.2, 0.4380156)) = 0.45685069 \text{ à } t_2 = 1.2$$

- Troisième itération

$$\hat{y} = y_2 + hf(t_2, y_2) = 0.45685069 + 0.1 \times f(1.2, 0.45685069) = 0.7217219$$

$$y_3 = y_2 + h/2(f(t_2, y_2) + f(t_2 + h, \hat{y}))$$

$$y_3 = 0.45685069 + 0.05 \times (f(1.2, 0.45685069) + f(1.3, 0.7217219)) = 0.74983045 \text{ à } t_3 = 1.3$$

5.3.2 Programme en Matlab de la méthode d'Euler modifier

```

    clc
    clear all
    close all
    a = input('la valeur minimale de l\'intervalle')
    b = input('la valeur maximale de l\'intervalle')
    n = input('le nombre de diviseure')
    x(1) = input('la valeure initiale')
    y(1) = input('la premiere valeure de la solution')
    h = (b - a)/n
    f = t * t - z * z + 1
    for i = 0 : n
    x(i + 1) = a + i * h
        end
    for i = 1 : n
    x1 = x(i) + h
    k1 = h * subs(f, t, x(i), z, y(i));
    k2 = h * subs(f, t, x1, z, y(i) + k1);
    y(i + 1) = y(i) + (h/2) * (k1 + k2);
        end
    y

```

5.4 La méthode de Runge-Kutta d'ordre 2

5.4.1 Introduction

Karl Rung (1856,1927) et Martin Kutta (1867,1944) on proposé en 1895 de résoudre le problème de Cauchy. Les méthodes Runge Kutta tire les avantages des méthodes de Taylor tout en gardant une Simplicité d'exécution de la méthode d'Euler. en pratique, Runge Kutta remplace l'évaluation analytique des ordres y^m , $m > 1$ par des dérivées numériques obtenues en évaluant la fonction $f(t, y(t))$ à différents endroits afin d'obtenir presque les mêmes résultats que ceux obtenu avec la méthode de Taylor.

5.4.2 Définition

Cette méthode est équivalente à la méthode d'Euler modifier, une méthode simple qui donner une solution approcher de la solution exacte de problème de Cauchy comme la méthode d'Euler, les méthodes de Runge Kutta peuvent être appliquées à une fonction arbitraire. La quasi-équivalence de la méthode Runge Kutta d'ordre 2 avec la méthode de

Taylor d'ordre 2 n'est pas évidente sans quelques manipulations mathématiques néanmoins, on peut rapidement vérifier que leur comportement au niveau numérique est similaire.

5.4.3 La formule générale de méthode de Runge-Kutta d'ordre 2

La formule de méthode de Runge Kutta d'ordre 2 s'écrit :

$$\begin{cases} y_{n+1} = y_n + ht^{(2)}(t_n, y_n) \\ y_{n+1} = y_n + h f(t_n + \frac{h}{2}, y_n + \frac{h}{2}f(t_n, y_n)) \end{cases} \quad (5.11)$$

Démonstration : La méthode de Runge-Kutta d'ordre 2 est donnée d'après le développement de Taylor d'ordre 2 suivante :

$$\begin{aligned} y_{n+1} &= y_n + hy'_n + \frac{h^2}{2}y''_n \\ &= y_n + h(y'_n + \frac{h}{2}y''_n) \\ &= y_n + ht^{(2)}(t_n, y_n) \end{aligned}$$

avec

$$\begin{aligned} t^{(2)}(t_n, y_n) &= y'_n + \frac{h}{2}y''_n \text{ est très minimale} \\ &= f(t_n, y_n) + \frac{h}{2}f'(t_n, y_n) \end{aligned}$$

le principe de la méthode de Runge-Kutta 2 est de remplacer $t^{(2)}(t_n, y_n)$ par $a_1 f(t + \alpha_1, y + \beta_1)$ où a_1, α_1, β_1 sont des constantes à déterminer. Celle-ci doivent être choisis de telle sorte que

$$t^{(2)}(t_n, y_n) = a_1 f(t + \alpha_1, y + \beta_1)$$

avec

$$y'_n = f(t_n, y_n), y''_n = f'(t_n, y_n)$$

et on a

$$df(t_n, y_n) = \frac{\partial f(t_n, y_n)}{\partial t} dt + \frac{\partial f(t_n, y_n)}{\partial y} dy$$

on peut écrire

$$\frac{df(t_n, y_n)}{dt} = f'(t_n, y_n) = \frac{\partial f(t_n, y_n)}{\partial t} + \frac{\partial f(t_n, y_n)}{\partial y} f(t_n, y_n)$$

en remplaçant dans l'expression de $t^{(2)}(t_n, y_n)$

$$\begin{aligned} t^{(2)}(t_n, y_n) &= f(t_n, y_n) + \frac{h}{2}f'(t_n, y_n) \\ &= f(t_n, y_n) + \frac{h}{2}\left(\frac{\partial f(t_n, y_n)}{\partial t} + \frac{\partial f(t_n, y_n)}{\partial y} f(t_n, y_n)\right) \end{aligned} \quad (5.12)$$

d'après la formule de Taylor applique à la fonction $f(t_n, y_n)$

$$f(t + \alpha_1, y + \beta_1) = a_1 \alpha_1 + \alpha_1 \frac{\partial f}{\partial t}(t, y) + \beta_1 \frac{\partial f}{\partial y}(t, y) + \frac{\alpha_1^2}{2!} + \frac{2\alpha_1 \beta_1}{2!} \frac{\partial^2 f(t, y)}{\partial t \partial y} + \frac{\beta_1^2}{2!} \frac{\partial^2 f(t, y)}{\partial y^2} \quad (5.13)$$

multiplions l'équation(5.13) par a_1 et comparer par rapport a l'équation (5.12)
par comparisont on obtient

$$\begin{cases} a_1 = 1 \\ a_1 \alpha_1 = \frac{h}{2} \\ a_1 \beta_1 = \frac{h}{2} f(t_n, y_n) \end{cases} \Leftrightarrow \begin{cases} a_1 = 1 \\ \alpha_1 = \frac{h}{2} \\ \beta_1 = \frac{h}{2} f(t_n, y_n) \end{cases} \quad (5.14)$$

avec l'erreur de la méthode de Runge-Kutta 2 est $e_2(t + \alpha_1, y + \beta_1) = \frac{\alpha_1^2}{2} \frac{\partial^2 f(t, y)}{\partial t \partial y} + \alpha_1 \beta_1 \frac{\partial^2 f(t, y)}{\partial t \partial y} + \frac{\beta_1^2}{2} \frac{\partial^2 f(t, y)}{\partial y^2}$
alors

$$\varepsilon_2(t + \frac{h}{2}, y + \frac{h}{2} f(t_n, y_n)) = \frac{h^2}{2} \frac{\partial^2 f(t, y)}{\partial t \partial y} + \frac{h^2}{4} f(t_n, y_n) \frac{\partial^2 f(t, y)}{\partial t \partial y} + \frac{h f(t_n, y_n)}{4} \frac{\partial^2 f(t, y)}{\partial y^2}$$

5.4.4 Précision de la méthode de Runge-Kutta 2

La méthode de Runge-Kutta 2 est une méthode du second ordre, c'est-à-dire que l'erreur au point t s'exprime par l'inégalité

$$|y_n - y(t_n)| \leq kh^2.$$

(La constante k est précisé dans la démonstration).

Démonstration : On procède $f \in C^2([t_0, t_0 + T])$ alors $y \in C^3([t_0, t_0 + T])$ et on a :

$$y(t_{n+1}) = y(t_n) + hy'(t_n) + \frac{h^2}{2} y''(t_n) + \frac{h^3}{6} y'''(\xi_n)$$

en soustrayant de la relation :

$$y_{n+1} = y_n + hf(t_n, y_n) + \frac{h^2}{2} \left(\frac{\partial f}{\partial t}(t_n, y_n) + \frac{\partial f}{\partial y}(t_n, y_n) f(t_n, y_n) \right) \quad (5.15)$$

On obtient :

$$e_{n+1} = e_n + h [f(t_n, y_n) - f(t_n, y(t_n))] + \frac{h^2}{2} \left[\begin{aligned} & \left(\frac{\partial f}{\partial t}(t_n, y_n) + \frac{\partial f}{\partial y}(t_n, y_n) f(t_n, y_n) \right) \\ & - \left(\frac{\partial f}{\partial t}(t_n, y(t_n)) + \frac{\partial f}{\partial y}(t_n, y(t_n)) f(t_n, y(t_n)) \right) \\ & - \frac{h^3}{6} y'''(\xi_n) \end{aligned} \right] \quad (5.16)$$

Comme $y \in C^3([t_0, t_0 + T])$ la dérive troisième est bornée, Posons :

$$M_3 = \max_{t \in [t_0, t_0 + T]} |y'''(t)|$$

$$|e_{n+1}| \leq |e_n| \left(1 + hL_0 + \frac{h^2}{2}L_1\right) + \frac{h^3}{6}M_3$$

et l'application des lemmes 1 et 2 donne le résultat avec :

$$k = \frac{h^2 M_3}{6L_0} \left[e^{((t_0+T)-t_0)(L_0 + \frac{h_0}{2}L_1)} - 1 \right]$$

5.4.5 Exercice d'application de la méthode de Runge-Kutta d'ordre 2

Exercice :

Faire trois itérations avec $h = 0.1$ par la méthode de Runge-Kutta d'ordre 2 pour l'équation différentielle suivante :

$$y'(t) = t^2 + y^2(t) + 1 \quad (y(1) = 0)$$

Solution :

$$\begin{cases} y(a) = y_1 \\ y_{n+1} = y_n + h f(t_n + \frac{h}{2}, y_n + \frac{h}{2}f(t_n, y_n)) \end{cases}$$

On a que $h = 0.1$, $t_0 = 1$, $y_0 = 0$ et que $f(t_n, y_n) = t_n^2 + y_n^2 + 1$.

– Pour la première itération, on obtient :

$$y_1 = y_0 + h f(t_0 + \frac{h}{2}, y_0 + \frac{h}{2}f(t_0, y_0)) = 0.1 \times ((1.05)^2 + (0.1)^2 + 1.) = 0.21125$$

alors $y_1 = 0.21125$

De même, on trouve que :

– Deuxième itération :

$$y_2 = y_1 + h f(t_1 + \frac{h}{2}, y_1 + \frac{h}{2}f(t_1, y_1)) = 0.21125 + 0.1 \times ((1.15)^2 + (0.324)^2 + 1.) = 0.454$$

alors $y_2 = 0.454$

– Troisième itération :

$$y_3 = y_2 + h f(t_2 + \frac{h}{2}, y_2 + \frac{h}{2}f(t_2, y_2)) = 0.454 + 0.1 \times ((1.25)^2 + (0.5863)^2 + 1.) = 0.7446$$

alors $y_3 = 0.7446$

5.4.6 Programme en Matlab de la méthode de Runge-Kutta d'ordre 2

```

clc
clear all
close all
a = input('la valeur minimale de l''intervalle')
b = input('la valeur maximale de l''intervalle')
n = input('le nombre des intervalles')
y(1) = input('la valeur initiale de la solution')
h = (b - a)/n
f = -z;
for i = 0 : n
x(i + 1) = a + i * h
end
for i = 1 : n
f1 = subs(f,t,x(i),z,y(i))
y(i + 1) = y(i) + f1 * h
end
x
y

```

5.5 La méthode de Runge-Kutta d'ordre 4

Une méthode simple qui donner une solution approcher de la solution exacte de problème de Cauchy. L'algorithme de Runge-Kutta d'ordre 4 repose sur une estimation plus précise de l'intégrale de l'équation et pour obtenir la formule de *Runge-Kutta d'ordre 4* on va utilise la même manière de l'obtenir de la méthode de *Runge-Kutta d'ordre 2*.

alors la formule de *Runge-Kutta d'ordre 4* est :

$$\begin{cases} y(t_{n+1}) = y(t_n) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) + \varepsilon_n(h^5). \\ y(t_0) = u_0 \end{cases} \quad (5.17)$$

avec

$$\begin{aligned} k_1 &= hf(t_n, y(t_n)) \\ k_2 &= hf(t_n + \frac{1}{2}h, y(t_n) + \frac{1}{2}k_1) \\ k_3 &= hf(t_n + \frac{1}{2}h, y(t_n) + \frac{1}{2}k_2) \\ k_4 &= hf(t_n + h, y(t_n) + k_3) \end{aligned}$$

L'idée est que la valeur suivante (y_{n+1}) est approchée par la somme de la valeur actuelle (y_n) et du produit de la taille de l'intervalle (h) par la pente estimée. La pente est obtenue par une moyenne pondérée de pentes

Notons que la pente est obtenue par une moyenne pondérée de pentes :

- k_1 est la pente au début de l'intervalle ;
- k_2 est la pente au milieu de l'intervalle, en utilisant la pente k_1 ;
- k_3 est la pente au milieu de l'intervalle en utilisant la pente k_2 ;
- k_4 est la pente à la fin de l'intervalle, en utilisant k_3 .

dans la moyenne des quatre pentes, un poids plus grand est donné aux pentes au point milieu.

5.5.1 Précision de la méthode de Runge-Kutta d'ordre 4

La méthode de Runge-Kutta d'ordre 4 est une méthode d'ordre quatre, c'est-à-dire que l'erreur au point t s'exprime par l'inégalité

$$|y_n - y(t_n)| \leq k h^4.$$

5.5.2 Exercice d'application de la méthode de Runge-Kutta d'ordre 4

Exercice :

Faire trois itérations avec $h = 0.1$ par la méthode de Runge-Kutta d'ordre 4 pour l'équation différentielle suivante :

$$y'(t) = t \sin(y(t)) \quad (y(0) = 2)$$

Solution :

$$\begin{cases} k_1 = h f(t_n, y_n) \\ k_2 = h f(t_n + h/2, y_n + k_1/2) \\ k_3 = h f(t_n + h/2, y_n + k_2/2) \\ k_4 = h f(t_n + h, y_n + k_3) \\ y_{n+1} = y_n + (1/6) * (k_1 + 2k_2 + 2k_3 + k_4) \\ t_{n+1} = t_n + h \end{cases}$$

On a que $h = 0.1, t_0 = 0, y_0 = 2$ et que $f(t_n, y_n) = t_n \sin(y_n)$.

donc $t_1 = t_0 + h = 0.1, t_2 = t_1 + h = 0.2, t_3 = t_2 + h = 0.3$.

– Pour la première itération, on obtient :

$$k_1 = h f(t_0, y_0) = 0.1 \times 0 \times \sin(2) = 0$$

$$k_2 = h f(0 + 0.05, 2 + 0) = 0.1 f(0.05, 2) = 0.1 \times 0.05 \times \sin(2) = 0.004546487$$

$$k_3 = 0.1 f(0.05, 2 + 0.004546487/2) = 0.1 f(0.05, 2.002273244)$$

$$= 0.1 \times 0.05 \times \sin(2.002273244) = 0.004541745$$

$$k_4 = 0.1 f(0.1, 2.004541745) = 0.00907398$$

alors

$$y_1 = 2 + (1/6)(k_1 + 2k_2 + 2k_3 + k_4) = 2.004541741.$$

De même, on trouve que :

– Deuxième itération :

$$k1 = 0.009074, k2 = 0.013582, k3 = 0.013568, k4 = 0.018032$$

$$y_2 = 2.01810947$$

– Troisième itération :

$$k1 = 0.018032, k2 = 0.022442, k3 = 0.022418, k4 = 0.026751$$

$$y_3 = 2.0405264$$

5.5.3 Programme en Matlab de la méthode de Runge-Kutta d'ordre 4

```

clc
clear all
close all
a = input('la valeur minimale de l\'intervalle')
b = input('la valeur maximale de l\'intervalle')
n = input('le nombre de diviseure')
x(1) = input('la valeur initiale')
y(1) = input('la premiere valeur de la solution')
h = (b - a)/n
f = t * t - z * z + 1
for i = 0 : n
x(i + 1) = a + i * h
end
for i = 1 : n
x1 = x(i) + h/2
x2 = x(i) + h
k1 = h * subs(f, t, x(i), z, y(i));
k2 = h * subs(f, t, x1, z, y(i) + k1/2);
k3 = h * subs(f, t, x1, z, y(i) + k2/2);
k4 = h * subs(f, t, x2, z, y(i) + k3);
y(i + 1) = y(i) + (1/6) * (k1 + 2 * k2 + 2 * k3 + k4);
end
y

```

5.6 Étude générale des méthodes a un pas

5.6.1 Consistance :

Définition "consistance"

l'erreur de consistance e_n relative à une solution exacte y est l'erreur $e_n = y(t_{n+1}) - y_{n+1}$; $\forall 0 \leq n < N$ en supposant $y_n = y(t_n)$. On a donc

$$e_n = y(t_{n+1}) - y_n - h_n \Phi(t_n, y_n, h_n)$$

on dit que la méthode est **consistante** si pour toute solution exacte y la somme des erreurs de consistance relatives à y , soit $\sum_{0 \leq n < N} |\varepsilon_n|$ tend vers 0, quand h_{\max} tend vers 0. C'est à dire

$$\lim_{h \rightarrow 0} \sum_{n=0}^N |\varepsilon_n| = 0.$$

Estimation de l'erreur de consistance

On a la méthode d'Euler est la formule suivant $y_{n+1} = y_n + hf(t_n, y_n)$, On suppose que la solution exacte vérifie $y \in C^2([t_0, t_0 + T], R)$

$$\varepsilon_n = y(t_{n+1}) - y_n - h_n f(t_n, y_n, h_n)$$

où $y(t_{n+1}) = y(t_n) + hy'(t_n) + \frac{h^2}{2}y''(\xi_n)$
 $y'(t_n) = f(t, y(t))$
 d'où

$$\varepsilon_n = \frac{h^2}{2}y''(\xi_n)$$

Majoration de l'erreur de consistance

On a l'erreur de consistance est $\varepsilon_n = \frac{h^2}{2}y''(\xi_n)$.

Majoration $M_2 = \sup_{[t_0, t_0+T]} |y''(t)| \Rightarrow \varepsilon_n \leq \frac{h^2}{2}M_2$.

Erreur due au schéma numérique

La solution exacte et le schéma numérique vérifient $y(t_{n+1}) = y(t_n) + hf(t_n, y(t_n)) + \varepsilon_n$
 $y_{n+1} = y_n + hf(t_n, y_n)$
 alors, comme $e_n = y(t_n) - y_n$, on obtient
 $e_{n+1} = e_n + h(f(t_n, y(t_n)) - f(t_n, y_n)) + \varepsilon_n$

Schéma de l'erreur de consistance

Thorme 5.1 Une condition nécessaire et suffisante pour que la méthode soit consistance avec l'équation différentielle est $\forall t \in [t_0, t_0 + T], \Phi(t, y, 0) = f(t, y)$.

Démonstration

Condition nécessaire La méthode est consistante, donc $\lim_{h \rightarrow 0} \max_n (\frac{1}{h}(y(t_{n+1}) - y(t_n)) - \Phi(t_n, y(t_n), h)) = 0$.
ou encore

$$\lim_{h \rightarrow 0} \max_n (\frac{1}{h} \int_{t_n}^{t_{n+1}} f(x, y(x)) dt - \Phi(t_n, y(t_n), h)) = 0.$$

pour tout $t \in [t_0, t_0 + T]$, il existe un encadrement $[t_n, t_{n+1}]$ telle que, même lorsque h tend vers 0. (i.e) lorsque n tend vers l'infini), $t \in [t_n, t_{n+1}]$. dans ces conditions, $\lim_{h \rightarrow 0} t_n = t$ et $\lim_{h \rightarrow 0} t_{n+1} = t$.

La continuité des fonctions f et Φ permet alors d'écrire

$$\begin{aligned} \lim_{h \rightarrow 0} \max_n (\frac{1}{h} \int_{t_n}^{t_{n+1}} f(x, y(x)) dt - \Phi(t_n, y(t_n), h)) &= f(t, y(t)) - \Phi(t, y(t), 0) = 0. \\ \Rightarrow f(t, y(t)) &= \Phi(t, y(t), 0). \end{aligned}$$

Condition suffisante de $f(t, y(t)) = \Phi(t, y(t), 0)$, il vient

$$\begin{aligned} &\frac{1}{h}(y(t_{n+1}) - y(t_n)) - \Phi(t_n, y(t_n), h) \\ &= \frac{1}{h} \int_{t_n}^{t_{n+1}} f(x, y(x)) dx - \Phi(t_n, y(t_n), h) \\ &= \frac{1}{h} \int_{t_n}^{t_{n+1}} (f(x, y(x)) - \Phi(t_n, y(t_n), h)) dx \\ &= \frac{1}{h} \int_{t_n}^{t_{n+1}} (\Phi(x, y(x), 0) - \Phi(t_n, y(t_n), h)) dx \end{aligned}$$

et, d'après la continuité de Φ ,

$$\lim_{h \rightarrow 0} \max_n (\Phi(x, y(x), 0) - \Phi(t_n, y(t_n), h)) = 0.$$

$$\text{puis } \lim_{h \rightarrow 0} \max_n (\frac{1}{h} \int_{t_n}^{t_{n+1}} f(x, y(x)) dt - \Phi(t_n, y(t_n), h)) = 0.$$

Définition "Ordre de méthode à un pas "

On dit qu'une méthode à 1 pas est d'ordre $\geq p$, si pour toute solution exacte y d'une équation différentielle où $y' = f(t, y)$ où f est de classe C^p , il existe une constante $C \geq 0$ telle que l'erreur de consistance relative à y vérifie

$$|\varepsilon_n| \leq Ch_n^{p+1}, \quad 0 \leq n < N.$$

Remarque La méthode d'Euler est une méthode d'ordre 1 car

$$\begin{aligned} y_{n+1} &= y_n + hy'(t_n) + \varepsilon_n(h^2) = y_n + hf(t_n, y_n) + \varepsilon_n(h^2) \\ &= y_{n+1} + \varepsilon_n(h^2). \end{aligned}$$

Consistance et ordre

Si $p \geq 1$ → la méthode à 1 pas est consistante.

5.6.2 La convergence

Définition “La convergence “

On dit que la méthode est convergente si pour toute solution exacte y , la suite (y_n) vérifie

$$\max_n |y_n - y(t_n)| \rightarrow 0$$

quand $y_0 \rightarrow y(0)$ et $h_{\max} \rightarrow 0$

Posons $\tilde{y}_n = y(t_n)$. Par définition d'erreur de consistance on a

$$\tilde{y}_{n+1} = \tilde{y}_n + h_n \Phi(t_n, \tilde{y}_n, h_n) + e_n.$$

Si la méthode est stable, de constante S , l'erreur globale est donc

$$\max_n |y_n - y(t_n)| \leq S(|y_0 - y(0)| + \sum_n |e_n|).$$

Démonstration de convergence de la méthode d'Euler :

Pour démontrer que la méthode d'Euler est convergente il faut démontrer que :

$$\varepsilon(h) = \lim_{h \rightarrow 0} \max_{0 \leq n \leq N} |e_n| \rightarrow 0$$

Soit la formule d'Euler $y_{n+1} = y_n + hf(t_n, y_n)$.

On a l'erreur de la méthode d'Euler est vérifiée la formule suivante :

$$|e_n| \leq e^{nLh} |e_0| + \frac{M}{2L} h(e^{nLh} - 1), 1 \leq n \leq N.$$

mais, pour $1 \leq n \leq N$, $nh \leq Nh = T$, et

$$|e_n| \leq e^{LT} |e_0| + \frac{M}{2} \frac{e^{LT} - 1}{L} h, \forall 1 \leq n \leq N,$$

ainsi, si $e_0 = 0$, $\varepsilon(h) \leq \frac{M}{2} \frac{e^{LT} - 1}{L} h$

et

$$\lim_{h \rightarrow 0} \varepsilon(h) = 0, \text{ donc } \lim_{h \rightarrow 0} \max_{0 \leq n \leq N} |e_n| \rightarrow 0$$

5.6.3 Stabilité :

Définition ”Stabilité” :

L'idée est de vérifier si de petites perturbations du schéma ne perturbent pas trop la solution approchée. On dit que la méthode est stable s'il existe une constante $S \geq 0$, telle que pour toutes suites (y_n) , (\tilde{y}_n) définies par

$$y_{n+1} = y_n + h_n \Phi(t_n, y_n, h_n); 0 \leq n < N$$

$$\tilde{y}_{n+1} = \tilde{y}_n + h_n \Phi(t_n, y_n, h_n) + \varepsilon_n; 0 \leq n < N$$

On ait

$$\max_n |\tilde{y}_n - y_n| \leq S(|\tilde{y}_0 - y_0| + \sum_n |\varepsilon_n|)$$

Théorème (CS de stabilité).

Pour que la méthode soit stable, il suffit que la fonction Φ soit Lipchitzienne en y , de constante L : $|\Phi(t, y_1, h) - \Phi(t, y_2, h)| \leq L |y_1 - y_2|$,

Pour tout $t \in [0, T]$, tout $(y_1, y_2) \in \mathbb{R}^2$ alors on peut prendre pour constante de stabilité

$$S = e^{\lambda t}.$$

Remarque :

L'intégration numérique est un vaste domaine d'analyse numérique, en générale ce n'est toujours évident d'intégrer des fonctions continues on reconnut à l'intégration numérique.

En général, un nombre de points de 4 à 6 est largement suffisant pour la plupart des applications.

Les méthodes de Gauss sont donne des résultats exactes pour les polynômes de degré inférieur ou égale au nombre des racines de polynôme qui choisit (le nombre des points).

5.7 Série des exercices

Exercice 5 - 1 :

Vérifier que les fonctions suivantes sont des fonctions Lipchitziennes en y sur $D \subset \mathbb{R}^2$.

- 1 - $f(x, y) = x |y|$, $D = \{(x, y) \in \mathbb{R}^2 / 0 \leq x \leq 2, -3 \leq y \leq 4\}$
- 2 - $f(x, y) = 1 + x \sin(x y)$, $D = \{(x, y) \in \mathbb{R}^2 / 0 \leq x \leq 2\}$,
- 3 - $f(x, y) = -y + x + 1$, $D = \{(x, y) \in \mathbb{R}^2 / 0 \leq x \leq 1\}$.

Exercice 5 - 2 :

01)- Démontrer que pour tout $T \neq 0$, ce problème admet une solution unique.

$$y' = 1 + x \sin(x y), \quad D = \{(x, y) \in \mathbb{R}^2 / 0 \leq x \leq T\} \text{ et } y(0) = 0.$$

02)- Démontrer que ce problème admet une solution unique.

$$y' = -y + \frac{x}{(1+x)^2}, \quad D \equiv \mathbb{R}^2 \text{ et } y(0) = 1.$$

Exercice 5 - 3 :

Utiliser la méthode d'Euler pour trouver les premières quatre valeurs de la solution $y = f(x)$ de l'équation différentielle

$$y' = \frac{y - x}{y + x}$$

qui satisfait la condition initiale $y(0) = 1$, en prenant le pas $h = 0.1$. Effectuer les calculs avec trois décimales exactes.

Exercice 5 - 4 :

Soit le problème de Cauchy suivant :

$$\begin{cases} y' = x - y + 1, & x \in [0, 1] \\ y(0) = 1, & h = 0.1 \end{cases}$$

Trouver les valeurs approchées de $y(x_i)$, $i = 0, 1, \dots, N$ par :

01)- la méthode d'Euler.

02)- la méthode d'Euler modifie.

Comparer les résultats obtenus avec la solution exacte.

Exercice 5 - 5 :

Calculer les valeurs approchées de la solution de l'équation différentielle

$$\begin{cases} y' = y - \frac{2x}{y} \\ y(0) = 1 \end{cases}$$

Aux points $x_1 = 0.2$ et $x_2 = 0.4$ d'après les méthodes Runge-Kutta d'ordre 2 et Runge-Kutta d'ordre 4. Effectuer les calculs avec 4 décimales exactes.

Estimer les résultats obtenus, si la solution exacte est $y(x) = \sqrt{2x+1}$.

Exercice 5 - 6 :

On considère, pour $t \in [0, 2]$, le problème de Cauchy suivant :

$$\begin{cases} y'(t) = \exp(-y(t)) \\ y(0) = 0 \end{cases} \quad (5.18)$$

- 1- Le problème admet-il une et une seule solution ?
- 2- Donner la solution exacte de ce problème. Quelle est la valeur de $y(2)$?
- 3- On prend un pas de temps $\Delta t = 0.5$. Quelle approximation de $y(2)$ obtient-on avec le schéma d'Euler ?

Exercice 5 - 7 :

On considère l'équation différentielle par :

$$\begin{aligned} y' &= -\frac{y}{(x+1)} \\ y(0) &= 1 \end{aligned}$$

avec $x \in [0, 1]$

- 1)- Montrer que cette équation admet une solution unique.
- 2)- Résoudre cette équation par la méthode de Runge-Kutta d'ordre 4 dans l'intervalle $[0, 1]$ avec un pas de 0.2.
- 3)- On considère l'équation différentielle par :

$$\begin{aligned} (1+x)y'' + y' &= 0 \\ y'(0) &= 1 \end{aligned}$$

avec $x \in [0, 1]$

Par un changement de variable adéquat, déduire de la question précédente les valeurs de : $y'(0.2)$, $y'(0.4)$, $y'(0.6)$, $y'(0.8)$ et $y'(1)$.

Bibliographie

- [1] François Liret. Maths en pratique, A L'usage des étudiants. Dunod, paris, 2006.
- [2] Jean-Pierre Demailly. Analyse numérique et équation différentielles. EDP-Sciences, 2006.
- [3] M. Lakrib. Cours d'Analyse Numérique. Office Publications Universitaires (Alger) 1995.
- [4] Jean-Étienne Rombaldi. Interpolation Approximation. Dépôt légal février 2005.
- [5] Kim Gaik Tay, Kian Boon Lim et all. Procedia-Social and Behavioral Sciences 90 (2013) 260-266.
- [6] André Fortin. Analyse numérique pour ingénieurs, Éditions de l'école polytechnique de Montréal, 2011.
- [7] Derradji Salah. Analyse numérique I, Office Publications Universitaires (Alger) 1990.
- [8] M. Fellah et N. H. Allal, Exercices corrigés en Analyse Numérique Elementaire, Office Publications Universitaires (Alger) 2005.