الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire

وزارة التعليم العالي و البحث العلمي
جامعة عمار ثليجي - الأغواط
كلية العلوم
العمادة

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Amar Télidji - Laghouat
Faculté des Sciences
Doyenné

المجلس العلمي

رقم : 05 / 2022

الأغواط في: 2022/04/26

مستخرج من محضر اجتماع المجلس العلمي
لكلية العلوم رقم 03 بتاريخ: 2022/04/26

- **المصادقة على المطبوعة البيداغوجية المقدمة من طرف الأستاذ بورقعة عبدالجبار:**

بناء على محضر اللجنة العلمية لقسم الرياضيات رقم: 04 المؤرخ في: 2022/04/24 و بعد الاطلاع على تقريري الخبيرين المختصين في دراسة هذه المطبوعة البيداغوجية - الأستاذين: **جمال وشنان- أستاذ محاضرأ- عبد العزيز رحمون - أستاذ محاضر أ- المعنونة ب:**

Méthodes Numériques et Programmation

للأستاذ: بورقعة عبدالجبار ، صادق المجلس العلمي للكلية على هذه المطبوعة .

رئيس المجلس العلمي لكلية العلوم

رئيس مجلس العلمي لكلية العلوم
السيد: ڤيباج جمال عبد الناصر

العنوان : ص.ب. رقم : 37 ج الأغواط
الهاتف: 029.14.53.00
الموقع : Mail : Http://www.lagh-univ.dz
الفاكس : 029.14.53.00

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire

وزارة التعليم العالي والبحث العلمي
جامعة عمار ثليجي بالاغواط
فرع التعليم المتلفزو التعليم عن بعد

Ministère de l'Enseignement Supérieuret de la Recherche Scientifique
Université Amar Telidji- Laghouat
Cellule de télé-enseignement et l'enseignement à distance

رقم :/72م أ ش ا ت م ت ع ب/2022 الاغواط: 2022/05/11

شهادة إدارية

أنا الممضي أسفله الدكتور شعيب نور الدين،مسؤول مركز الأنظمة وشبكاتالإعلام و الاتصال و التعليم المتلفزو التعليم عن بعد، أصرّح بأن:

السيد:بورقعة عبد الجبار،

أستاذ: كلية العلوم ،

قد قام بنشر عمل بيداغوجي على موقع جامعة عمار ثليجي -الاغواط للتعليم عن بعدelearning.lagh-univ.dz، مصادق عليه في محضر المجلس العلمي لكلية العلومرقم03 المنعقد بتاريخ2022/04/26، كما هو موضح في الجدول الآتي:

المقياس	نوع العمل البيداغوجي		المستوى	الرابطعبرالخط
	مطبوعة (PDF ou PPT)	دروسعبرالخط (سمعي أو بصري)		
Méthodes Numériques et Programmation	PDF	-	السنة الثانية ليسانس	http://elearning.lagh-univ.dz/course/view.php?id=5358

الإمضاء

مسؤول مركز الانظمة وشبكة الإعلام والاتصال والتعليم المتلفز والتعليم عن بعد بجامعة الاغواط
إمضاء: شعيب نور الدين

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
Université Amar Telidji de Laghouat
Faculté des Sciences
Département de Sciences de la Matière

Polycopié de Cours

Intitulé:

Méthodes Numériques et Programmation

Semestre (3) - 2ème année - SM

Présentée par:

BOUREGA Abdeldjabar

Email: jabourega@gmail.com

Année Universitaire: 2021-2022.

Avant-propos

Pour une première version, de cet ouvrage, je compte sur votre indulgence et vous invite à nous faire part de vos remarques, commentaires ainsi que vos critiqués par le biais de ma boite mail jabourega@gmail.com et soyez sur de ma gratitude et mon parfond respect.

BOUREGA Abdeldjabar

Programme Pédagogique

$2^{ème}$ année Domaine Sciences de la Matière
Filière « Physique -Chimie »
Information Licence - Deuxième Année : Physique -Chimie
Département des Sciences de la Matière
Semestre : S3
Unité : UEM3
Crédit : 3
Coefficient: 2
Contenu de la matière:

Chapitre 1. Initiation (ou rappel) de langages de programmation informatique MATLAB et/ou MATHEMATICA et/ou FORTRAN et/ou C^{++}

Chapitre 2. Intégration numérique
2. 1 Méthode des Trapèzes
2. 2 Méthode de Simpson

Chapitre 3. Résolution numérique des équations non-linéaires
3. 1 Méthode de Bissection
3. 2 Méthode de Newton

Chapitre 4. Résolution numérique des équations différentielles ordinaires
4. 1 Méthode d'Euler
4. 2 Méthode de Runge-Kutta

Chapitre 5. Résolution numérique des systèmes d'équations linéaires
5. 1 Méthode de Gauss
5. 2 Méthode de Gauss-Seidel

Table des matières

Introduction

L'objectif de cette polycopie est de premette à l'étudiant l'acquisition de cette matière et lui faciliter la matièr de l'outil numérique par la compréhension des langages de programmation évolués d'une part, et d'autre part, par l'utilisation des méthodes numériques de résolution de systèmes d'équations algébriques.

La maîtrise de l'outil numérique par l'enseignement des langages de programmation évolués d'une part, et d'autre part, par l'enseignement des méthodes numériques de résolution de systèmes d'équations algébriques.

Les étudiants(es) en science possèdent souvent des connaissances mathématiques très développées, néanmoins il a été constaté qu'ils trouvent des difficultés à concrétiser ces connaissances sur un ordinateur. La rédaction de cette polycopie de cours s'inscrit dans cette optique, afn de mettre à la disposition des étudiants(es), d'outils pratiques aidant à la stimulation de leurs connaissances opérationnelles. Cette polycopie s'adresse à tous les étudiants(es) suivant un cursus universitaire de type scientifique, à l'instar de la physique, la chimie, la biologie, filières technologiques...etc. Les prérequis exigés sont relatifs aux notions élémentaires en mathématique appliquée, abordées durant les premières années du cycle universitaire. Bien évidemment, la liste des méthodes numériques présentées ici est strictement conformes au programme officiel.

Cette polycopie s'articule autour de cinq chapitres. Dans le premier chapitre est initiation (ou rappel) de langages de programmation informatique MATLAB. Le second chapitre est consacré à l'intégration numériques (méthode du trapèze et celle de Simpson). Dans le troisième chapitre, traite la recherche de racines d'une fonction réelle de variable réelle (méthode de dichotomie, Newton-Raphson). Le quatrième chapitre mis en lumière les diverses techniques de résolution numériques d'équations différentielles (méthode de Euler, méthode de Runge-Kutta). Enfn, le cinquième chapitre résolution numérique des systèmes d'équations linéaires (Méthode de Gauss, Méthode de Gauss-Seidel).

Chapitre 1

Rappels sur les langages informatiques initiation a Matlab

1.1 Introduction à MATLAB

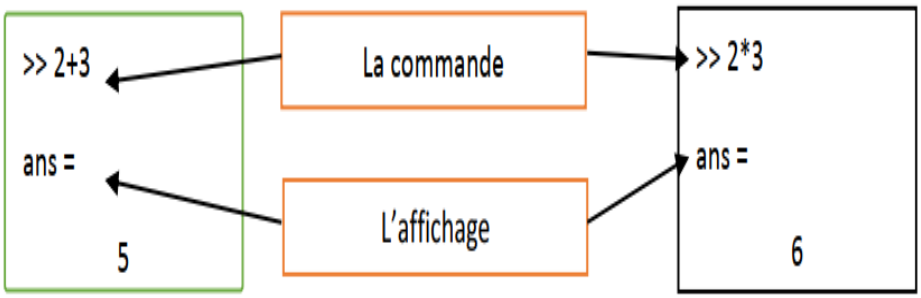
MATLAB est un logiciel de calcul et de visualisation, dont les entités de base sont des matrices: MATLAB est une abréviation de Matrix Laboratory.
MATLAB est un langage interprété, il propose des facilités de programmation et de visualisation, ainsi qu'un grand nombre de fonctions réalisant diverses méthodes numériques.
MATLAB est un langage de programmation, de haut niveau, intégrant plusieurs fonctionnalités, il fournit un environnement interactif pour la création et la gestion de programmes. Dans ce qui suit on introduit quelques fonctionnalités de bases qui nous permettrons d'apprendre à manipuler le logiciel et de nous familiariser avec le langage.
L'élément fondamental chez MATLAB est la matrice, puisque le type de base des données est le type « array ». Scalaires, vecteurs, matrices réelles et complexes sont considérés comme étant des cas spéciaux du type de base.

1.1.1 Quelques bases de l'interface Matlab

1. Le symbole [>>] : dans Le Workspace (MATLAB) indique où entrer les commandes

2. Le mot (ans) (réponse) désigne le résultat de l'opération effectuée dans Le Workspace (MATLAB).

Exemple 1.1.1



Les variables et les opération (+, -, *, /)

- **Les opérations de base**: Ceux-ci agissent sur les scalaires et les matrices. Dans le cas matriciel, les dimensions des matrices doivent s'y prêter!

Symbole	Description	Exemple
$+, -, *, /$	Les opérations de base en mathématiques (addition, soustraction, multiplication et division)	$7+9$ $3/4$
p_i	La constante p_i	$p_i/3$
cos sin tan	Les fonctions trigonométriques usuelles	$cos(3*p_i/2)$
log exp	Le logarithme népérien et l'exponentielle	$exp(3)$
sqrt	La racine carrée	$sqrt(5)$
^	La puissance	4^5

Opérations de base sur les matrices

Matlab propose tout un ensemble d'opérations usuelles sur les matrices

Symbole	Description	Exemple
$+ - *.$	Les opérations de base(addition, soustraction, produit matriciel)	$A + B$
	Les tailles des matrices doivent être compatibles	A^*B
^	La puissance matricielle (itération du produit matriciel)	$A\hat{}3$
'	Transposée d'une matrice	A'
inv	L'inversion d'une matrice (sison inverse existe)	$inv(A)$

D'autres opération peuvent être réalisées sur chaque élément de A

Symbole	Description	Exemple
$+ - */.$	Réalise l'opération entre un scalaire et chaque élémentde la matrice	$5.4 * A$
$.*$	Réalise la multiplication terme à terme de deux matrices de même taille.	$A. * B$
.^	Met à une certaine puissance chaque élément de la matrice	$A.\hat{}3$

Opérateurs relationnels

Initiation au logiciel MATLAB

$<$	Strictement inférieur
$<=$	Inférieur ou égal
$>$	Strictement supérieur
$>=$	Supérieur ou égal
$==$	Égal
$\tilde{} =$	Différent

Remarque 1.1.2 *La division à gauche $A\backslash B$ donne le résultat de l'opération AB^{-1} (utilisée pour résoudre un système linéaire).*

1.2 La structure conditinnal (If)

Structures de contrôle

MATLAB dispose des instructions de contrôle suivantes : if, switch, for et while. La syntaxe de chacune de ces instructions est semblable à celles des langages classiques. Il est important de savoir que beaucoup d'opérations nécessitent ces instructions dans des langages classiques tels que C et Fortran, alors que dans MATLAB, on peut s'en affranchir. Les opérations sur les matrices (addition, multiplication, etc.) sont les exemples les plus évidents.

Instructions alternatives

- L'instruction: if - elseif - else

La syntaxe est la suivante :

```
if(test)
        commandes
else
        Autres commandes
end
```

On peut également imbriquer des if... else les uns dans les autres à l'aide de l'instruction elseif.

Exemple 1.2.1 *vérifier si un entier naturel donné n est pair ou impair*

if rem$(n, 2)$
disp('nombre pair')
else
disp('nombre impair')
and

Remarque 1.2.2 *La fonction rem : retourne le reste de la division de deux nombres.*

Exemple 1.2.3 *La valeur absolue*

Lire

if $x > 0$

$x = x$

else

$x = -x$

end

Ecrive $(''la\ valeur\ absolu\ '', x)$

1.3 Les boucles (for, while)

1.3.1 Boucles et tests

Les principales instructions de contrôle proposées par MATLAB sont for, while et if; elles fonctionnent à peu près comme leurs équivalents dans les autres langages de programmation.

La boucle for

La boucle for, doit respecter la syntaxe suivante :

for compteur = expression

instructions

end

Generalement, expression est un vecteur de la forme début :incrément :fin et compteur prend successivement toutes les valeurs de expression pour éxecuter instructions.

Exemple 1.3.1 *Les instructions suivantes permettent de calculer une valeur approchée de* e^x*, pour* $x = 10$*, en utilisant la formule:*

$$e^x \simeq \sum_{k=0}^{n} \frac{x^k}{k!}$$

$>> x = 10;$

$>> n = 50;$

$>> s = 1;$

$>> terme = 1;$

$>> for\ k = 1 : n,\ terme = x^{*} terme\ /k; s = s + terme;\ end;$

$>> s$

Exemple 1.3.2 *Calculer, à l'aide de la boucle for, la somme des n premiers entiers naturels.*

La boucle while

L'instruction while respecte la syntaxe suivante:

while expression

instructions

end

expression désigne le résultat d'une operation logique. Elle est construite en utilisant des opérateurs relationnels et logiques.

La boucle while répète un bloc d'instructions tant qu'une condition donnée est vraie

Exemple 1.3.3 *.Les instructions suivantes ont le même effet que les précédentes:*

$>> f(1) = 0;\ f(2) = 1;\ k = 3;$

$>> while\ k <= 6$

$f(k) = f(k-1) + f(k-2); k = k + 1;$

end

1.4 Les tableaux (veteure et matrice)

1.4.1 Vecteurs et matrices

La structure de données de Matlab est le tableau ; même un nombre est considéré comme une matrice 1×1. Toutes les fonctions et opérations relatives aux tableaux sont très optimisées et sont à utiliser aussi souvent que possible.

Création

Un tableau (Matrice) est délimité par des crochets. On sépare les colonnes par des espaces et les lignes par des points-virgules. Exemple

$\rangle\rangle A = [123; 456]$ $\rangle\rangle B =$

$A =$ $B = [1; 2; 3]$

$$\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{matrix} \qquad \begin{matrix} 1 \\ 2 \\ 3 \end{matrix}$$

Opération sur les matrices

Addition / soustraction: 2 matrices de mêmes dimensions (même nombre de lignes et de colonnes) s'additionnent ou se soustraient terme à terme.

Exemple 1.4.1

$$\begin{aligned} A &= [1\ 2\ 3; 4\ 5\ 6], \qquad B = [2\ 4\ 6; 7\ 8\ -1] \\ A + B &= [3\ 6\ 9; 11\ 13\ 5] \\ A - B &= [-1\ -2\ -3; -3\ -3\ 7] \end{aligned}$$

Multiplication ou division: la multiplication ou division de matrice obéit à des règles spéciales qu'on ne détaille pas ici. En revanche on peut multiplier ou diviser 2 matrices terme à terme par l'emploi des symboles « .∗» ou « ./»

Exemple 1.4.2

$$\begin{aligned} A &= [123; 456], \quad B = [2\ 4\ 6; 7\ 8\ -1] \\ A.^*B &= [2818; 2840 - 6] \\ A./B &= [0.50.50.5; 0.570.62 - 6] \end{aligned}$$

On ra donner quelques commandes de base et fonction sur les matrices dans Matlab

1.4.2 Commandes de base dans Matlab

Commande	Description
ones(n, m)	Matrice de taille $n \times\ m$ ne contenant que des 1.
zeros(n, m)	Matrice de taille $n \times\ m$ ne contenant que des 0.
eye(n, m)	Matrice de taille $n \times\ m$ contenant des 1 sur la première diagonale et des 0 ailleurs.
rand(n, m)	Matrice de taille $n \times\ m$ contenant des nombres aléatoires.
diag(v)	Matrice diagonale où les éléments de la diagonale sont les composantes du vecteur v.

1.4.3 Fonctions sur les matrices

Commande	Description
$det(A)$	Renvoie le déterminant de A ; celle-ci doit être carrée.
trace(A)	Renvoie la trace de A.
rank(A)	Renvoie le rang de A
null(A)	Renvoie une base du noyau de A; l'argument supplémentaire $'r'$donne une «meilleure »
diag(A)	Renvoie la première diagonale de A.
norm(v)	Renvoie la norme euclidienne de v; v est un vecteur. Il est aussi possible de calculer d'a
mean(A)	Renvoie une liste contenant la moyenne des éléments de chaque colonne.
sum(A)	Renvoie une liste contenant la somme des éléments de chaque colonne.
prod(A)	Renvoie une liste contenant le produit des éléments de chaque colonne.
$max(A)$	Renvoie une liste contenant la valeur maximale chaque colonne.
$min(A)$	Renvoie une liste contenant la valeur minimale de chaque colonne.
length(A)	Renvoie le maximum entre le nombre de lignes et de colonnes ;

1.4.4 Instructions de contrôle

break	Termine l'exécution d'une boucle
continue	Passe à l'itération suivante en sautant les instructions suivantes dans une boucle
else	Sinon, utilisé avec if
elseif	Sinon si, utilisé avec if
end	Termine for, if et while
error(.)	Termine un programme et retourne un message d'erreur
for	Répétition
if	Instruction conditionnelle
otherwise	Sinon, utilisé avec switch
pause	Arrêt momentané d'un programme (attente d'appui sur une touche)
return	Retour à la fonction appelante ou au clavier
switch	Branchement conditionnel
while	Tant que

1.4.5 Instructions mathématiques de base

Sqrt(.)	Racine carrée	abs(.)	Valeur absolue ou module
exp(.)	Exponentielle	angle(.)	Phase
log(.)	Logarithme naturel	conj(.)	Conjugaison complexe
log10(.)	Logarithme décimal	imag(.)	Partie imaginaire
log2(.)	Logarithme de base 2	Isreal(.)	Vrai si réel
nextpow2(.)	Puissance de 2 supér	real(.)	Partie réelle
cos(.)	Cosinus	fix(.)	Arrondi vers zéro
acos(.)	Arc cosinus	floor(.)	Arrondi vers $-\infty$
sin(.)	Sinus	ceil(.)	Arrondi vers $+\infty$
asin(.)	Arc sinus	round(.)	Arrondi au plus proche entier
cot,tan(.)	(Co)Tangente	sign(.)	Signe
atan(.)	Arc tangente	factorial(.)	Factorielle
sinc(.)	Sinus	cardinal rem(.)	Reste division euclidienne
cosh,sinh, tanh(.)	Fonctions hyperboliques	mod(.)	Reste modulo

1.4.6 Principaux symboles mathématiques reconnus par MATLAB.

minuscules grecques		majuscules grecques		symboles mathematiques	
commande	symbole	commande	symbole	commande	symbole
$\backslash apla$	α	$\backslash Delta$	Δ	$\backslash leq$	$\leq$
$\backslash beta$	β	$\backslash Theta$	θ	$\backslash geq$	$\geq$
$\backslash gamma$	γ	$\backslash Gamma$	Γ	$\backslash int$	$\int$
$\backslash omega$	ω	$\backslash Omega$	Ω	$\backslash \inf ty$	∞
$\backslash lambda$	λ	$\backslash Lambda$	Λ	$\backslash sim$	$\sim$
$\backslash pi$	π	$\backslash Phi$	Φ	$\backslash partial$	∂
$\backslash rho$	ρ	$\backslash Psi$	Ψ	$\backslash in$	$\in$
$\backslash nu$	υ	$\backslash \mathrm{Si}\, qma$	Σ	$\backslash rightar \mathrm{row}$	$\rightarrow$

1.5 Les polynômes

MATLAB représente un polynôme sous forme d'un tableau de ses coefficients classés dans l'ordre des puissances décroissantes.

Saisie d'un polynôme: Le polynôme P d'expression: $P(x) = 2x^2+3x-5$, est représenté par le tableau à une dimension suivant :

$>> P = [2\ 3 - 5]$

$P =$

$2\ 3 - 5$

Racines d'un polynôme : On peut déterminer les racines des polynômes P à l'aide de la fonction root.

Exemple 1.5.1 $>> roots(P)$

$ans =$

$1.0000 + 2.0000i$

$1.0000 - 2.0000i$

1.5.1 Représentation graphique

Pour tracer la représentation graphique du polynôme $Q(x)$, définissons un domaine pour la variable x qui contient les racines de Q.

Exemple 1.5.2 $>> x = 0 : 0.1 : 3;$

$>> y =polyval(Q, x);$

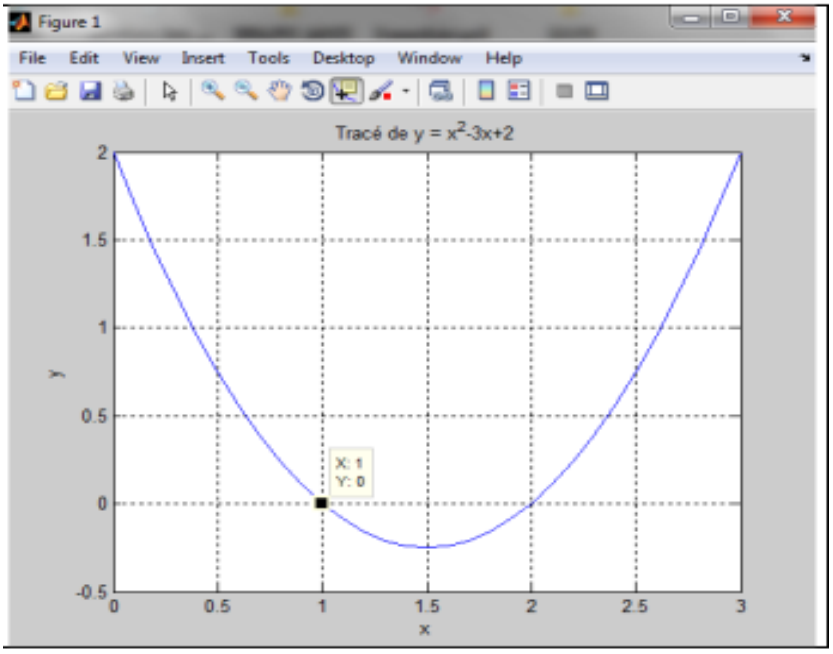
$>> plot(x, y)$

$>>grid$

$>>title\ ('Tracéde\ y = x\hat{}2 - 3x + 2')$

$>> xlabel('x')$

$>> ylabel('y')$



1.6 Plotage graphe fonction

1.6.1 Tracer des courbes dans MATLAB

Pour tracer n'importe quelle courbe dans Matlab, une ou deux variables ou plus doivent être présentes. Pour cela, les variables doivent être déclarées, puis après on donne la commande $plot(x, y)$.

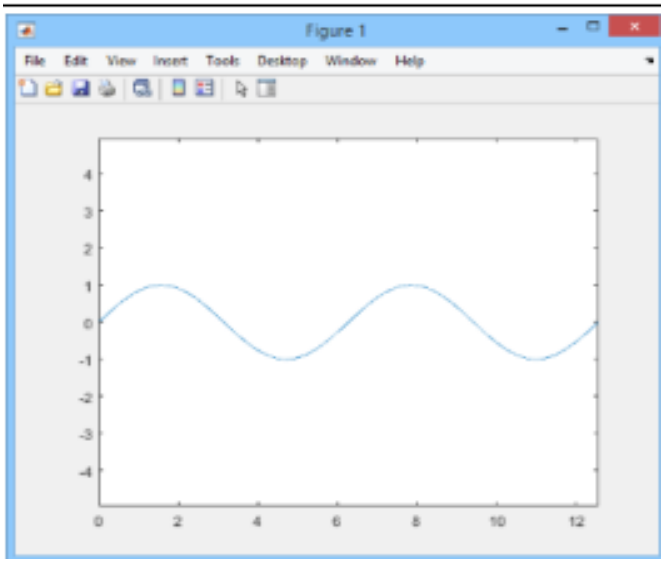
Donc, si nous voulons tracer y en fonction de x, MATLAB connecte les points de coordonnées $(x(k), y(k))$ de $1 \leq k \leq$longueur(x).

En prenant un grand nombre de points dans le vecteur x puis en définissant $y = f(x)$ pour une fonction f, le graphique de la fonction (x, y) nous donnera le graphique de la fonction.

Exemple 1.6.1 $>> x = [0 : 0.01 : 4^*pi];$

$>> y = sin(x);$

$>> plot(x, y),\ axis\ equal$



Chapitre 2

Intégration numérique

Lorsque le calcul analytique est difficile (voir impossible) le calcul numérique de l'intégral d'une fonction s'impose. L'une des manières les plus naturelles pour calculer l'aire sous une courbe est d'approximer par une aire facilement calculable.

Dans les méthodes d'intégration, l'intégrale d'une fonction continue sur un intervalle borne $[a, b]$ 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 composées dans lesquelles la fonction est remplacée par un polynôme d'interpolation sur chaque intervalle élémentaire $[x_i; x_{i+1}]$ de la subdivision, et les méthodes de Gauss fondées sur les polynômes orthogonaux pour lesquelles les points de la subdivision sont imposés.

Position du problème: L'objectif de l'intégration numérique est de calculer l'intégrale I d'une fonction $f(x)$ sur un certain intervalle $[a; b]$.

On veut évaluer l'intégrale d'une fonction f sur un intervalle $[a, b]$. Si l'on connait sa primitive F, alors

$$I = \int_a^b f(x)\, dx = F(b) - F(a).$$

Mais dans de nombreux cas, F ne peut pas être connue. Ou notons que l'expression

analytique de $f(x)$ peut être connue comme elle peut être inconnue.

Exemples 2.0.2 : $\int_a^b e^{-x^2}dx, \quad \int_a^b \frac{\sin(x)}{x}dx, \quad \int_a^b \sqrt{\sin(x)}dx.$

2.1 Méthode des trapèzes

2.1.1 Méthode des trapèzes simple

Pour évaluer numériquement $I(f) = \int_a^b f(x)\,dx$, on divise l'intervalle borne $[a;b]$ en n parties $[x_0;x_1]$, $[x_1;x_2]$, ..., $[x_{n-1};x_n]$ de même longueur $h = \frac{b-a}{n}$ et on considère les points d'intégration

$$x_0 = a,\ x_1 = x_0 + h, ...,\ x_i = x_0 + ih\ (i = 0, ..., n),\ x_n = b.$$

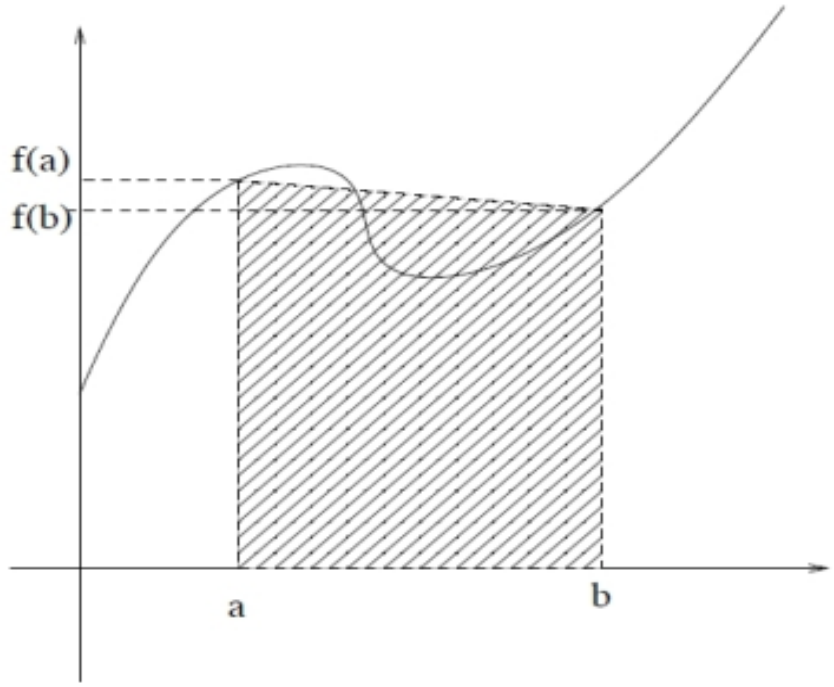


Figure1-Méthode des trapèze simple

On considère cette fois l'interpolation par un polynôme de degré un construit sur les points a et b. On obtient la formule classique des trapèzes:

$$\int_a^b f(x)\,dx \simeq \frac{h}{2}\left(f(a)+f(b)\right).$$

C'est la formule simple des trapèzes sur l'intervalle $[a,b]$.

$$S = (Petite\_base + Grande\_base) \times \frac{Hauteur}{2}$$

Petite base et grande base correspondent à $f(a)$ et $f(b)$ et Hauteur$= h$ $(h = b-a)$

Exemples 2.1.1 $I = \int_a^b x dx = \frac{b-a}{2}(a+b)$

$J = \int_0^1 e^{-x^2} dx \simeq \frac{1-0}{2}\left(e^{-0^2} + e^{-1^2}\right) \simeq 0,6845$

$k = \int_{\frac{\pi}{2}}^{\pi} \frac{\sin(x)}{x} dx \simeq \frac{\pi-\frac{\pi}{2}}{2}\left(\frac{\sin\left(\frac{\pi}{2}\right)}{\frac{\pi}{2}} + \frac{\sin(\pi)}{\pi}\right) \simeq \frac{\pi}{4}\left(\frac{2}{\pi}+0\right) = \frac{1}{2}.$

2.1.2 Méthode des trapèzes généralisés

L'aire $I(f)$ comprise entre $[a;b]$ et le graphe de f peut être approchée par la somme des aires des trapèzes induits par les points $x_0, x_1, ..., x_n$. Sur chaque intervalle $[x_i, x_{i+1}]$; $0 \leq i \leq n-1$:

$$\int_{x_i}^{x_{i+1}} f(x)\,dx \simeq \frac{x_{i+1}-x_i}{2}\left(f(x_i)+f(x_{i+1})\right).$$

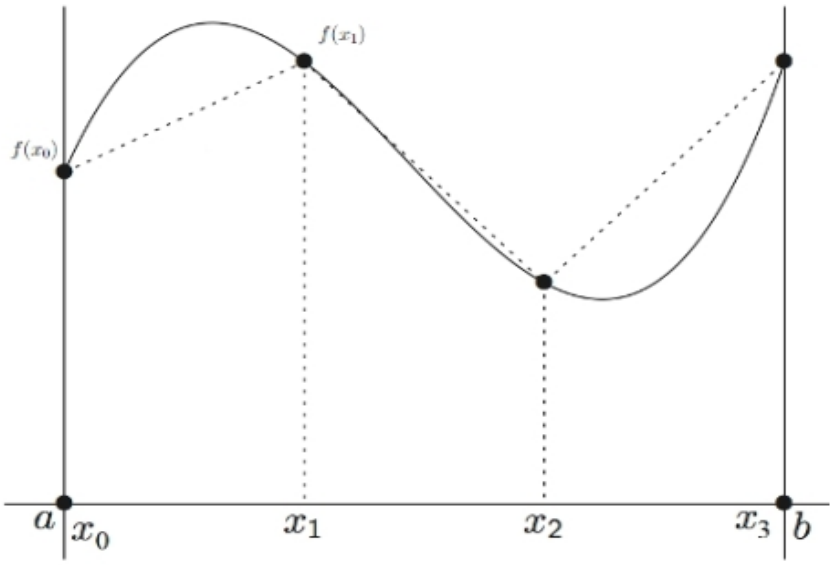


Figure 2- Méthode des trapèze généralisés

$$\begin{aligned}
I(f) &= \int_a^b f(x)\,dx = \int_{x_0=a}^{x_1} f(x)\,dx + \int_{x_1}^{x_2} f(x)\,dx + ... + \int_{x_{n-1}}^{x_n=b} f(x)\,dx \\
&= \sum_{i=0}^{n-1} \int_{xi}^{x_{i+1}} f(x)\,dx \\
&= \frac{h}{2} \sum_{i=0}^{n-1} (f(x_i) + f(x_{i+1})) \\
&= \frac{h}{2} (f(x_0) + f(x_1) + f(x_1) + f(x_2) + f(x_2) + f(x_3) + ... + f(x_{n-1}) + f(x_n)) \\
&= \frac{h}{2} (f(x_0) + 2f(x_1) + 2f(x_2) + 2f(x_3) + ... + 2f(x_{n-1}) + f(x_n))
\end{aligned}$$

D'où

$$I_n(f) = \frac{h}{2} \left(f(a) + f(b) + 2 \sum_{i=1}^{n-1} f(x_i) \right).$$

C'est la formule des trapèze composite sur l'intervalle $[a; b]$.

Exemple 2.1.2 *On calcule l'intégrale suivante*

$$I_4 = \int_1^3 \ln(x)\, dx.$$

Donner une valeur approchée de l'intégrale I_4 en utilisant la méthode des trapèzes composite avec 4 sous-intervalles. Cela revient à prendre $h = \frac{1}{2}$ et $x_0 = a = 1$, $x_4 = b = 3$, $x_1 = \frac{3}{2}$, $x_2 = 2$ et $x_3 = \frac{5}{2}$. D'après la méthode des trapèzes, on a:

$$I_4 = \frac{h}{2}\left(f(1) + f(3) + 2\left(f\left(\frac{3}{2}\right) + f(2) + f\left(\frac{5}{2}\right)\right)\right) \simeq 1,821.$$

2.1.3 Recherche de l'erreur d'approximation si $f \in C^2([a;b])$ Méthode des trapèzes Simple

Soit $f : [a;b] \to \mathbb{R}$ continue. Posons

$$E_1 = I(f) - I_1(f) = \int_a^b f(x)\, dx - \frac{b-a}{2}\left(f(a) + f(b)\right).$$

f'' est continue sur $[a;b]$, alors d'après le théorème des valeurs intermédiaires, $\exists\ \theta_1 \in [a;b]$:

$$E_1 = \frac{-h^3}{12} f''(\theta) = \frac{-(b-a)^3}{12} f''(\theta).$$

E_1 l'erreur de la méthoder $n = 1$. Implique

$$|E_1| \leq \frac{(b-a)^3}{12} M_2.$$

Où $M_2 = max_{\theta \in [a;b]} |f''(\theta)|$.
la formule simple des trapèzes sur l'intervalle $[a;b]$.

$$I(f) = \frac{b-a}{2}\left(f(a) + f(b)\right) - \frac{(b-a)^3}{12} f''(\theta).$$

Exemple 2.1.3 $I = \int_0^1 e^{-2x} dx$,

Soit $f(x) = e^{-2x}$, $a = 0$; $b = 1$; $f''(x) = 4e^{-2x}$; $f^{(3)}(x) = 8e^{-2x} < 0$.

Parce que $f^{(3)}(x) < 0$ *donc* f'' *est décroissant strictement*

Où $M_2 = max_{x\in[0;1]} |f''(x)| = f''(0) = 4$

$$|E_1| < \frac{(1-0)^3}{12}4 = \frac{1}{3} \simeq 0,333, \quad \varepsilon \simeq 0,333.$$

Remarque 2.1.4 *la formule, à deux points, des trapèzes est clairement si* f *est un polynôme de degré inférieur ou égal à* 1.

De plus, cette formule est de degré de précision égal à 1.

En effet, pour $f(x) = x^2$; $n = 1$; $x_0 = a$; $x_1 = b$, *on a:*

$$\begin{aligned} I &= \int_a^b x^2 dx = \left[\frac{x^3}{3}\right]_a^b = \frac{b^3}{3} - \frac{a^3}{3} = \frac{1}{3}(b-a)\left(b^2+ab+a^2\right) \\ I_1 &= \frac{b-a}{2}\left(b^2+a^2\right). \end{aligned}$$

On voit clairement que $I \neq I_1$.

2.1.4 Recherche de l'erreur d'approximation si $f \in C^2([a;b])$ Méthode des trapèzes composite

Soit $f : [a;b] \to \mathbb{R}$ continue de max M et min m. Alors, $\forall y \in [m;M]$, $\exists x \in [a;b]$ / $f(x) = y$. Posons

$$E_n = I(f) - I_n(f) = \int_a^b f(x)\,dx - \frac{b-a}{2}\left(f(a) + f(b) + 2\sum_{i=0}^{n-1} f(x_i)\right).$$

avec $x_i = x_0 + ih$ $(h = \frac{b-a}{n})i = 0, 1, ..., n$. f'' est continue sur $[a;b]$, alors d'après le théorème des valeurs intermédiaires, $\exists\theta \in [a;b]$:

$$E_n = -\frac{nh^3}{12}f''(\theta) = -\frac{n\frac{(b-a)^3}{n^3}}{12}f''(\theta) = -\frac{(b-a)^3}{12n^2}f''(\theta).$$

Où $M_2 = max_{\theta\in[a;b]} |f''(\theta)|$. Comme, on obtient

$$|E_n| \leq \frac{(b-a)^3}{12n^2} M_2$$

la formule composite des trapèzes sur l'intervalle $[a;b]$:

$$\exists \theta \in [a;b] \ / \ \ I(f) = \frac{b-a}{2}\left(f(a) + f(b) + 2\sum_{i=0}^{n-1} f(x_i)\right) - \frac{(b-a)^3}{12n^2} f''(\theta).$$

Exemple 2.1.5 *: On considère l'intégrale*

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

Quel nombre de sous-intervalles n faut-il choisir pour avoir une erreur inférieure à 10^{-4}.

L'erreur est majorée par

$$|E_n| \leq \frac{(b-a)^3}{12n^2} M_2 \leq \varepsilon$$

Où $M_2 = max_{\theta\in[1;2]} |f''(\theta| = max_{\theta\in[1;2]} \frac{2}{\theta^3} = 2$. Donc ici on a

$$|E_n| \leq \frac{(2-1)^3}{12n^2}.2$$

Pour que $|E_n| < 10^{-4}$ il faut que

$$\frac{1}{6n^2} \leq 10^{-4}$$

i.e. $n > \frac{102}{\sqrt{6}} \simeq 40,8$. A partir de $n_0 = [40,8] + 1 = 40 + 1 = 41$ sous-intervalles, l'erreur de quadrature est inférieure à 10^{-4}.

2.2 Méthode de Simpson

2.2.1 Méthode de Simpson simple

Dans le cadre de cette méthode, on interpole chaque 3 points $(a; f(a)), (\frac{a+b}{2}; f(\frac{a+b}{2})), (b; f(b))$. Comme trois points induisent deux subdivisions, le nombre $n = 2$ de subdivisions doit être pris pair $(n = 2m;\ m = 1)$.

C'est la première formule de Simpson simple sur l'intervalle $[a, b]$. D'où

$$\int_a^b f(x)\,dx \simeq \frac{h}{3}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right), \quad h = \frac{b-a}{2}.$$

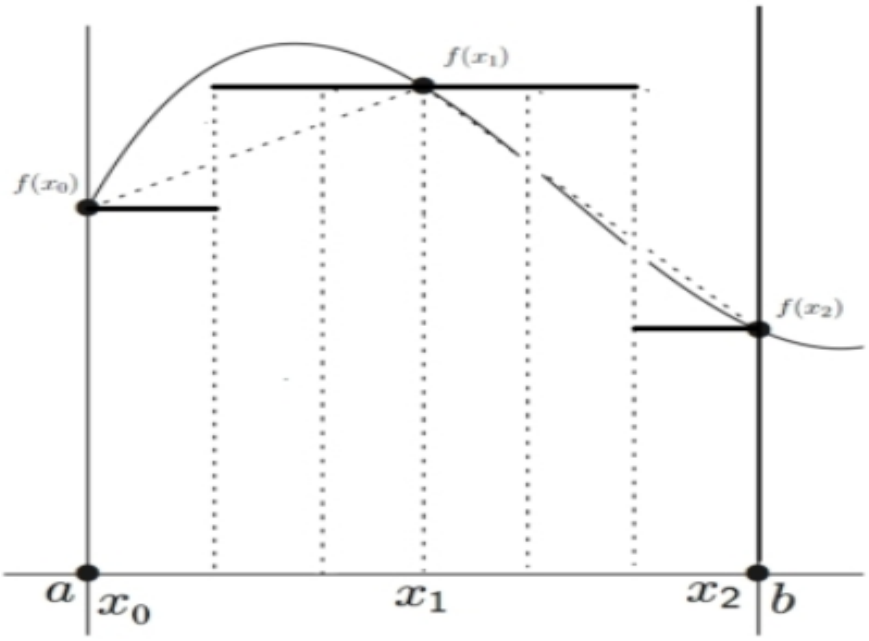


Figure 3-Méthode de Simpson simple

Exemple 2.2.1 *Calculer à l'aide de la méthode des Simpson l'intégrale*

$I = \int_0^{\pi} \sin x^2 dx = \frac{\frac{\pi-0}{2}}{3}\left(\sin 0^2 + 4\sin\left(\frac{0+\pi}{2}\right)^2 + \sin \pi^2\right) = \frac{\pi}{6}\left(0 + 4\sin\frac{\pi^2}{4} + \sin \pi^2\right) \simeq -0,095$

2.2.2 Méthode de Simpson généralisés

Sur chaque intervalle $[x_{2i}; x_{2i+2}]$; $0 \leq i \leq n-1$:

$$\int_{x_{2i}}^{x_{2i+2}} f(x)dx \simeq \frac{h}{3}\left(y_{2i} + 4y_{2i+1} + y_{2i+2}\right); \quad y_i = f(x_i);$$

Donc

$$\begin{aligned} I(f) &= \int_a^b f(x)\,dx = \int_{x_0=a}^{x_2} f(x)\,dx + \int_{x_2}^{x_4} f(x)\,dx + ... + \int_{x_{2m-2}}^{x_{2m}=b} f(x)\,dx \\ &= \frac{h}{3}\left(y_0 + 4y_1 + y_2 + y_2 + 4y_3 + y_4 + y_4 + 4y_5 + y_6 + ... + y_{2m-2} + 4y_{2m-1} + y_{2m}\right) \\ &= \frac{h}{3}\left(y_0 + 4\left(y_1 + y_3 + ... + y_{2m-1}\right) + 2\left(y_2 + y_4 + y_6 + ... + y_{2m-2} + y_{2m}\right)\right) \\ &= \frac{h}{3}\left[y_0 + y_n + 4\sum_{(i\ \ impaire)} y_i + 2\sum_{(i\ \ paire)} y_i\right] \end{aligned}$$

D'où

$$I_n(f) = \frac{h}{3}\left[y_0 + y_n + 4\sum_{(i\ \ impaire)} y_i + 2\sum_{(i\ \ paire)} y_i\right] \quad \text{où} \quad h = \frac{b-a}{n}$$

Or

$$I_n(f) = \frac{h}{3}\left[f(x_0) + f(x_n) + 4\sum_{i=1}^{m-1} f(x_{2i-1}) + 2\sum_{i=0}^{m-2} f(x_{2i+2})\right] \quad \text{où} \quad m = \frac{n}{2}.$$

C'est la première de Simpson composite sur l'intervalle $[a; b]$.

Exemple 2.2.2 *On veut calculer l'intégrale* $I = \int_0^6 x^3 dx$. *Pour* $n = 2$ *on a* $h = 3$, *d'où*

$$I = \frac{h}{3}(f(0) + f(6) + 4f(3)) = 324.$$

Pour $n = 4$ *on a* $h = \frac{3}{2}$, *d'où*

$$I = \frac{h}{3}(f(0) + f(6) + 2f(3) + 4(f(\frac{3}{2}) + f(\frac{9}{2})) = 324.$$

D'autre part, la valeur exacte de cette intégrale est

$$I = \int_0^6 x^3 dx = \frac{x^4}{4}]_0^6 = 324.$$

On remarque que la méthode de Simpson est exacte pour le polynôme de degré inférieur ou égal à 3.

Remarque 2.2.3 *Ceci implique que la formule de Simpson est exacte si f est un polynôme de degré inférieur ou égal à* 3. *De plus, cette formule est de degré de précision égal à* 3.
En effet ; pour $f(x) = x^4; n = 2; x_0 = a;\ x_1 = \frac{a+b}{2}$ *et* $x_2 = b$

$$I = \int_a^b x^4 dx = \left[\frac{x^5}{5}\right]_a^b = \frac{b^5}{5} - \frac{a^5}{5} = \frac{1}{5}(b-a)\left(b^4 + ab^3 + a^2b^2 + a^3b + a^4\right).$$

et

$$I_2 = \frac{\frac{b-a}{2}}{3}\left(a^4 + b^4 + 4\left(\frac{a+b}{2}\right)^2\right) = \frac{b-a}{24}\left(5b^4 + 4ab^3 + 6a^2b^2 + 4a^3b + 5a^4\right).$$

On voit clairement que $I \neq I_2$.

2.2.3 Recherche de l'erreur d'approximation si $f \in C^4([a;b])$

On rappelle que l'erreur associée la méthode des **Simpson simple** s'écrit, si $f \in C^4([a;b])$,

$$\exists \zeta \in [a;b] \ \ / \ \ E_2 = I(f) - I_2(f) = -\frac{h^5}{90} f^{(4)}(\zeta) = -\frac{\frac{(b-a)^5}{2^5}}{90} f^{(4)}(\zeta) = -\frac{(b-a)^5}{2880} f^{(4)}(\zeta);$$

On rappelle que l'erreur associée la méthode des **Simpson généralisés** s'écrit, si $f \in C^4([a;b])$,

$$\exists \zeta \in [a;b] \ \ / \ \ E_n = I(f) - I_n(f) = -\frac{nh^5}{180} f^{(4)}(\zeta) = -\frac{n\frac{(b-a)^5}{n^5}}{180} f^{(4)}(\zeta) = -\frac{(b-a)^5}{180n^4} f^{(4)}(\zeta);$$

Par consequent

$$|E_n| \leq \frac{(b-a)^5}{180n^4} M_4 \leq \varepsilon.$$

Où $M_4 = max_{\zeta \in [a;b]} \left| f^{(4)}(\zeta) \right|$.

Exemple 2.2.4 *On considère l'intégrale* $I = \int_2^3 \frac{1}{2x+1} dx$

1− *Donner une majoration de l'erreur commise cas es Simpson simple.*

2− *Quel nombre de sous-intervalles n faut-il choisir pour avoir une erreur inférieure à* 10^{-6}.

− *L'erreur est majorée par*

$$|E_n| \leq \frac{(b-a)^5}{180n^4} M_4 \leq \varepsilon.$$

Parce que Simpson simple donc $n = 2$

$$f'(\zeta) = -\frac{2}{(2\zeta+1)^2};\ f'(\zeta) = \frac{8}{(2\zeta+1)^3}; f^{(3)}(\zeta) = -\frac{48}{(2\zeta+1)^4};\ f^{(4)}(\zeta) = \frac{288}{(2\zeta+1)^5}.$$

Parce que $f^{(5)}(\zeta) = -\frac{2304}{(2\zeta+1)^6} < 0$, $f^{(4)}(\zeta)$ *donc est décroissant* $f^{(4)}$ *est positive alors*

$$M_4 = max_{\zeta \in [2;3]} \left| f^{(4)}(\zeta) \right| = \left| f^{(4)}(2) \right| = \frac{288}{(2 \times 2 + 1)^5} = \frac{288}{3125} \simeq 0,09216;$$

Donc ici on a:

$$|E_2| \leq \frac{(3-2)^5}{2880} \times 0,09216 \simeq 0,000032.$$

2− *Pour que* $|E_n| < 10^{-6}$ *il faut que*

$$\frac{(3-2)^5}{180n^4} \times 0,09216 \leq 10^{-6}.$$

i.e. $n > 4,7568$. *A partir de* $n_0 = [4,7568] + 1 = 4 + 1 = 5$ *sous-intervalles, l'erreur de quadrature est inférieure à* 10^{-6}.

Exemple 2.2.5 *Calculons l'intégrale* $I = \int_0^1 e^{-x^2} dx$. *Utilisons les méthodes élémentaires précédentes à l'aide des valeurs de* $f(x) = e^{-x^2}$ *aux points* $x_0 = 0$; $x_1 = \frac{1}{2}$ *et* $x_2 = 1$.

Alors on a

$$f(0) = 1; \;\; f(\frac{1}{2}) \simeq 0,7880 \quad et \quad f(1) \simeq 0,36788.$$

calculons l'erreur

$$f'(x) = -2xe^{-x^2}; \;\; f''(x) = 2(2x^2-1)e^{-x^2}, \quad f'''(x) = 4x(-2x^2+3)e^{-x^2} \geq 0 \quad sur \quad [0;1].$$

donc f'' *est croissant sur* $[0;1]$.

$$M_2 = max_{x\in[0;1]} |f''(x)| = |f''(0)| = 2.$$

Méthode des trapèzes pour $n = 1$

$$I = \int_0^1 e^{-x^2} dx = \frac{h}{2}(f(0) + f(1)) \simeq 0,68394,$$

avec une erreur

$$|E_2| \leq \frac{M_2}{12}(b-a)^3 = \frac{2}{12}(1-0)^3 \simeq 0,16666.$$

Méthode de Simpson pour $n = 2$

$$\int_0^1 e^{-x^2} dx \simeq \frac{h}{6}(f(0) + f(1) + 4f(\frac{1}{2})) \simeq 0,74718,$$

avec une erreur

$f^{(4)}(x) = 4(4x^4 - 12x^2 + 3)e^{-x^2} > 0 \quad sur \quad [0;1].,\quad f^{(4)}(x) = 8x(-4x^4 + 2x^2 - 12x - 3)e^{-x^2} < 0 \; sur \;\; [0;1].$

$M_4 = max_{\zeta\in[2;3]} \left|f^{(4)}(\zeta)\right| = \left|f^{(4)}(0)\right| = 12$

$$|E_2| \leq \frac{M_4}{2880}(b-a)^5 = \frac{12}{2880}(1-0)^5 \simeq 0,004166.$$

;;

La valeur exacte est $I_{exacte} = 0,74682$. *On remarque que* $E = |I_{exacte} - I_S| = 0,00036 \simeq$

4×10^{-4}, *donc la méthode de Simpson donne une approximation à* 4×10^{-4} *avec seulement trois valeurs de* f *parce que les dérivées d'ordre supérieur ne varient pas trop sur l'intervalle* $[0; 1]$.

La formule d'intégration numérique de Trapèze

L'algorithme

$\Delta x = \frac{b-a}{n}$

for $k = 1$ to $(n-1)$

$x = a + \Delta x \times k$

$s = s + f(x)$

and for

$T = \Delta x \frac{f(a)+f(b)}{2} + \Delta x \times s$

Aid

1. les algorithmes des formules doivent être implémentés dans des procédures séparées (**Trapèze et Simpson**) dans deux programmes.

2. la fonction $f(x)$ à intégrer doit être implémentés dans une fonction séparées (f).

3. le programme principal, pour chaque formule, doit servir pour:

a.lire les bornes de l'intervalle d'intégation $[a, b]$.

b.lire le nombre de division N.

c. et doit servir pour afficher la vleur de l'intégrale: T (Trapèze) et S (Simpson)

d. et doit afficher l'erreur d'approximation(valeur calculée-valeur exacte). la valeur exacte doit être calculé analytiquement.

Application

Ecrire un programme MATLAB qui implémente la méthode numérique d'intégration (**formule de Trapèze**) et applique cette méthode pour trouver l'intégration de la fonction suivante avec un nombre de division $N = 10$:

$$\int_1^2 \left(3x^2 + 2x\right) dx.$$

Afficher l'erreur d'approximation aprés le calcule de la valeur exacte.

clc

clear

f=@$(x)\, 3 * x.2 + 2 * x$;

h=0,1;

a=1; b=2;

ep=0,001;

$y = f\,(x)$ $y = [5 \;\; 5.83 \;\; 6.72 \;\; 6.72 \;\; 6.72 \;\; 6.67 \;\; 8.68 \; 9.95 \;\; 10.88 \;\; 2.07 \;\; 13.32 \;\; 14.63 \;\; 16]$

$I = MyTrapezes\text{Fun}(f, a, b, eps)$

$N = length\,(x)$-1;

h=$(x\,(N+1) - x\,(N))\,/N$; $h = (b-a)\,/N$

I=0;

for k=1;N

I=I+y$(k) + y\,(k+1)$

end

I=h/2*I

end

I=

10,0049999999999

La formule d'intégration numérique de Simpson

La formule de Simpson est une amélioration du calcule de surface dans une amélioration de approximation de l'intégrale car elle utilise des formes de parabole (polynôme de degré (2)) pour approximer les courbes des fonctions traitées**.** La formule de Simpson

$$S = \frac{\Delta x}{3}\left[y_0 + 4y_1 + 2y_2 + 4y_3 + 2y_4 + ... + 4y_{n-2} + 2y_{n-1} + y_n\right] \qquad \Delta x = \frac{b-a}{n}.$$

L'algorithme

$\Delta x = \frac{b-a}{n}$;

$h = \frac{\Delta x}{2}$;

for $k = 1$ à n

$x = a + h \times (2k - 1)$

$s_1 = s_1 + f(x)$

and for

for $k = 1$ à $(n - 1)$

$x = a + h \times 2k$

$s_2 = s_2 + f(x)$

and for

$S = h \times \frac{(f(a)+f(b)+4s_1+2s_2)}{3}$;

Application

Ecrire un programme MATLAB qui implémente la méthode numérique d'intégration (**formule de Simpson**) et applique cette méthode pour trouver l'intégration de la fonction suivante avec un nombre de division $N = 5$:

$$\int_1^2 \left(3x^2 + 2x\right) dx.$$

Afficher l'erreur d'approximation aprés le calcul de la valeur exacte.

Programme principal

clc, clear

f=@$(x)\, 3 * x.2 + 2 * x$;

x=1:0,1:2;

y=f(x)

$I = My\,\mathrm{Si}\,mpson\mathrm{Fun}(x, y)$

Sous-Programme

fonction $I = My\,\mathrm{Si}\,mpson\mathrm{Fun}(x, y)$

Sn=0;

$n = length\,(x)$-1;

mn=mod$(n, 2)$

x=ones$(3, 1)$

x=ones(3, 1)

if mm =0

m =0

for j=n-1:n+1

m=m+1;

y(m) =y(j) ;

for k=1:3

$x\left(m,k\right)=x\left(j\right)\widehat{}\left(3-k\right)$

end

end

a=X/Y

$S_1=a\left(1\right)*\left(x\left(n+1\right)\widehat{}3-x\left(n\right)\wedge 3\right)/3$

$S_2=a\left(2\right)*\left(x\left(n+1\right)\widehat{}3-x\left(n\right)\wedge 3\right)/2$

$S_3=a\left(3\right)*\left(x\left(n+1\right)\widehat{}1-x\left(n\right)\wedge 3\right)/1$

$S_n=S_1+S_2+S_3$;

n=n-1;

end

$s=zeros\left(1,n\right)$

for i=2:2:n

m=0

for j=i-1:i+1

m=m+1

Y(m) =Y(j) ;

for k=1:3

X(m,k) =X$(m,k)\widehat{}(3-K)$:

end

end

a=X/Y;

$S_1 = a\,(1) * (x\,(n+1)\widehat{}3 - x\,(n) \wedge 3)\,/3$

$S_2 = a\,(2) * (x\,(n+1)\widehat{}3 - x\,(n) \wedge 3)\,/2$

$S_3 = a\,(3) * (x\,(n+1)\widehat{}1 - x\,(n) \wedge 3)\,/1$

$S\,(i) = S_1 + S_2 + S_3;$

end

I=sum$(s) + s_n;$

end

I=

10

Chapitre 3

Résolution numérique des équations non-linéaires

Position du problème

Soit $f : [a, b] \subset \mathbb{R} \to \mathbb{R}$ une fonction au moins continue sur $[a, b]$. On cherche les racines de f c'est à dire les points $x \in [a, b]$ tels que $f(x) = 0$.
Si $f(a).f(b) < 0$, alors par le théorème des valeurs intermédiaires, il existe au moins une racine de f dans l'intervalle $[a, b]$. Si de plus, f est strictement monotone alors f est une bijection et cette racine est unique dans $[a, b]$.
Nous supposerons l'unique racine de f dans l'intervalle $[a, b]$.
En général, les méthodes de résolution numérique de $f(x) = 0$ sont des méthodes itératives.
Elles consistent à construire une suite $(x_n)_n$ convergente (le plus rapidement possible) vers α.

Séparation des racines

La séparation des racines s'effectue, en général, en utilisant deux types de méthodes:

Méthode graphique:

Soit on trace (expérimentalement ou par étude de variations de f) le graphe de la fonction f et on cherche son intersection avec l'axe (Ox).
Soit on décompose f en deux fonctions f_1 et f_2 simples à étudier, telles que $f = f_1 - f_2$, et on cherche les points d'intersection des graphes de f_1 et f_2, dont les abscisses sont exactement les racines de f.

Remarque 3.0.6 *On choisit souvent f_1 et f_2 de façon que leurs courbes soient des courbes connues.*

Exemple 3.0.7 *La fonction f définie par: $f(x) = xe^x - 1$, $x \in]0, +\infty[$ a une racine unique dans l'intervalle $[0, 1]$. En effet, posons $f_1(x) = e^x$ et $f_2(x) = \frac{1}{x}$.*
Alors $f(x) = 0 \Longleftrightarrow f_1(x) = f_2(x)$ et d'après le graphe

$$G(f_1) \cap G(f_2) = \{(\alpha, f(\alpha)), \quad \alpha \in]0, 1[\},$$

Méthode de balayage:

On considère une suite croissante finie $\{x_i, \quad i = 0, 1, ..., n\}$ de valeurs de x réparties sur l'intervalle $[a, b]$. Si f est continue et $f(x_i).f(x_{i+1}) < 0$, alors il existe entre x_i et x_{i+1} au moins une racine de f (c'est le théorème des valeurs intermédiaires).

Exemple 3.0.8 *Le polynôme P défini par: $P(x) = x^4 - 6x - 7$ a au moins deux racines réelles $\alpha_1 \in]-2, 0[$, $\alpha_2 \in]2, 3[$ car: $P(-3) = 56; .., p(-2) = 21$, $P(0) = -7$, $P(2) = -3$, $P(3) = 56$ et $P(4) = 225$,..., ect.*

3.1 Méthode de dichotomie (ou de la bissection)

Cette méthode est utilisée pour approcher les racines d'une fonction continue $f : \mathbb{R} \to \mathbb{R}$. S'il existe a, b $(a < b)$ avec $f(a).f(b) < 0$, on sait alors qu'il existe au moins une racine de f dans l'intervalle $]a, b[$.

Posons $a_0 = a$, $b_0 = b$, $I_0 =]a_0, b_0[$ et $x_0 = \frac{a_0+b_0}{2}$.

Pour $n \geq 1$, on choisit le sous-intervalle $I_n =]a_n, b_n[$ de l'intervalle $]a_{n-1}, b_{n-1}[$ de la façon suivante:

a) posons $x_{n-1} = \frac{a_{n-1}+b_{n-1}}{2}$

b) si $f(x_{n-1}) = 0$, alors $\alpha = x_{n-1}$ et la méthode est terminée,

c) si $f(x_{n-1}) \neq 0$, alors

i) si $f(a_{n-1}).f(x_{n-1}) < 0$, alors $\alpha \in]a_{n-1}, x_{n-1}[$ et on pose $a_n = a_{n-1}$, $b_n = x_{n-1}$,

ii) si$f(x_{n-1}).f(b_{n-1}) < 0$, alors $\alpha \in]x_{n-1}, b_{n-1}[$ et on pose $a_n = x_{n-1}$, $b_n = b_{n-1}$,

d) on définit l'intervalle $I_n =]a_n, b_n[$, on augmente n de 1 et on recommence du point a).

On obtient donc une suite de valeurs approchées de α.

En répétant (itérant) la même méthode pour l'intervalle obtenu on aura les valeurs :

$$x_0 = \frac{a_0 + b_0}{2},\ x_1 = \frac{a_1 + b_1}{2}, ..., x_n = \frac{a_n + b_n}{2}.$$

La suite $\{x_n\}_{n=0,+\infty}$ converge vers la solution $\overline{x}$ de $f(\overline{x}) = 0$ lorsque $n \to +\infty$.

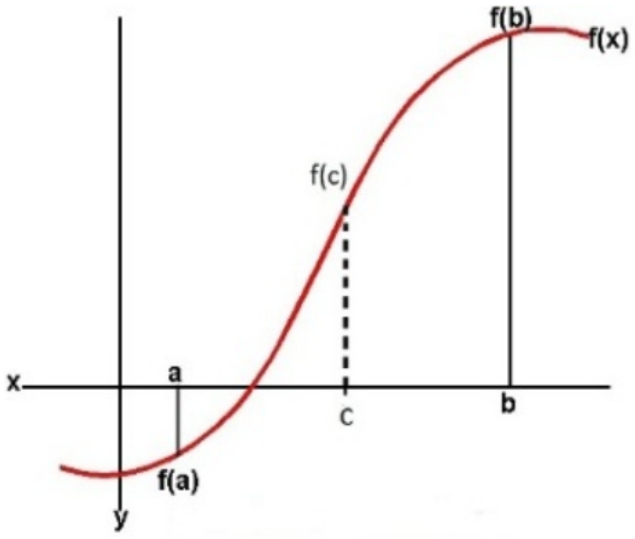


Figure 4- Méthode de Bissection

Nombre de divisions pour avoir une précision ε donné.

Puisque chaque fois on divise l'intervalle en deux parties égales, on a :

$$b_0-a_0=\frac{b-a}{1},\quad b_1-a_1=\frac{b-a}{2},\quad b_2-a_2=\frac{b_1-a_1}{2}=\frac{\frac{b-a}{2}}{2}=\frac{b-a}{2^2},...,\quad b_n-a_n=\frac{b-a}{2^n}.$$

Puisque $\alpha\in[a_n,b_n]=[a_n,x_n]\cup[x_n,b_n]$, on a , $|x_n-\alpha|\leq\frac{b_n-a_n}{2}=\frac{\frac{b-a}{2^n}}{2}=\frac{b-a}{2^{n+1}}$.
Il faut que la différence $|x_n-\alpha|$ qui est l'erreur du calcul soit inférieure à une précision donnée ε, c'est-à-dire:

$$|x_n-\alpha|\leq\varepsilon$$

Alors, il suffit que

$$\frac{b-a}{2^{n+1}}\leq\varepsilon$$

Cela donne

$$n\geq\frac{ln\left(\frac{b-a}{2\varepsilon}\right)}{ln2}. \tag{3.1}$$

Remarque 3.1.1 1−*L'inégalité* (4.1) *nous permet d'estimer le nombre suffisant d'itérations* n *pour approcher avec une précision donnée* ε, *il su ra de résoudre l'inégalité:* $\frac{b-a}{2^{n+1}}<\varepsilon$ *par rapport à* n. *Il suffira de prendre*

$$n_0=\left[\frac{ln\left(\frac{b-a}{2\varepsilon}\right)}{ln2}\right]+1.$$

2−*Cette méthode est inconditionnellement convergente, son problème c'est qu'elle est lente c'est pourquoi elle est utilisée pour démarrer d'autres méthodes plus élaborées.*

Exemple 3.1.2 *Soit l'equation* $x^3+2x-7=0$. *Donné la solution approximer utiliser la méthode de bissection pour approximer cette solution au rang de* 0.005.
On a: 1. f *est continue*
2. $f(1).f(2)<0$
3.$f'(x)=3x^2+2>0$ *donc* f *est monotone.*

Alors par le théorème des valeurs intermédiaires, il existe unique solution dans l'intervalle $[a,b]$. *déte* $|x_n - \alpha| \leq \frac{b-a}{2^{n+1}} \leq \varepsilon$

$\frac{2-1}{2^{n+1}} \leq 0.005 \Longrightarrow 10^2 \leq 2^n \Longrightarrow n \geq \frac{\ln\left(10^2\right)}{\ln(2)} \Longrightarrow n_0 = \left[\frac{2\ln(10)}{\ln(2)}\right] + 1 = 6 + 1 = 7$

n	a_n	b_n	$x_n = \frac{a_n+b_n}{2}$	$f(x_n)$	$\lvert x_n - x_{n-1}\rvert$
0	1	2	$1,5$	$f(1,5) < 0$	/
1	$1,5$	2	$1,75$	$f(1,75) < 0$	$\lvert 1,75 - 1,5\rvert = 0,25 > 0.005$
2	$1,5$	$1,75$	$1,625$	$f(1,625) < 0$	$\lvert 1,625 - 1,75\rvert = 0,125 > 0.005$
3	$1,5$	$1,625$	$1,5625$	$f(1,5625) < 0$	$\lvert 1,5625 - 1,625\rvert = 0,0625 > 0.005$
4	$1,5625$	$1,625$	$1,59375$	$f(1,59375) < 0$	$\lvert 1,59375 - 1,5625\rvert = 0,03125 > 0.005$
5	$1,5625$	$1,59375$	$1,578$	$f(1,578) < 0$	$\lvert 1,578 - 1,59375\rvert = 0,00675 > 0.005$
6	$1,5625$	$1,578125$	$1,5703$	$f(1,5703) < 0$	$\lvert 1,5703 - 1,578\rvert = 0,0077 > 0.005$
7	$1,5625$	$1,5703$	$1,5664$	$f(1,5664) < 0$	$\lvert 1,5664 - 1,5703\rvert = 0,0039 < 0.005$

Donc la solution $\alpha = 1,5664 \bar{+} 0.005$.

3.2 Méthode de Newton-Raphson (méthode de la tangente)

Supposons qu'on a déterminé un intervalle $[a;b]$ dans lequel f admet une racine séparée.

Interprétation géométrique

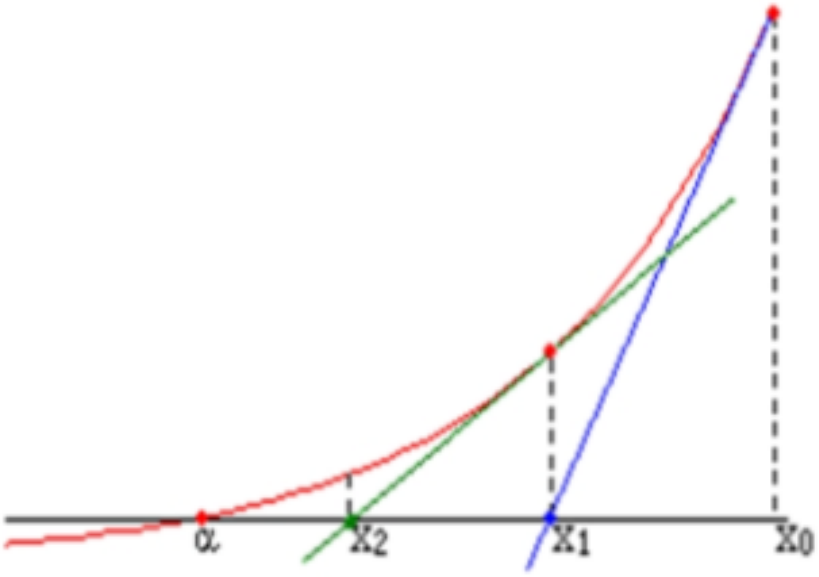


Figure 4-Méthode de la tangente

Graphiquement la méthode de Newton-Raphson fonctionne comme suit: à partir d'un point x_0 bien choisi dans $[a,b]$, x_1 est l'abscisse du point d'intersection de la tangente de graphe de f au point $(x_0, f(x_0))$ avec l'axe des abscisses. D'où

$$f'(x_0) = \frac{0 - f(x_0)}{x_1 - x_0} = \frac{-f(x_0)}{x_1 - x_0} \Longrightarrow x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} (\text{ si } f'(x_0) \neq 0)$$

En répétant le processus sur x_1, on obtient un point x_2, et ainsi de suite.
Les points $(x_n)_{n\in N}$ vérifient donc la relation de récurrence:

$$\begin{cases} x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \\ x_0 \in [a,b] \end{cases}$$

c'est la formule de Newton-Raphson la plus utilisée dans la recherche des racines de f.

Théorème 3.2.1 *Soit $f \in C^2([a,b])$ vérifiant:*

1.$f(a).f(b) < 0$;

2.$f'(x) \neq 0$, $\forall x \in [a,b]$, *(c.à.d:* f' *ne change pas de signe dans* $[a,b]$;

3. $f''(x) \neq 0, \forall x \in [a,b]$ *est de signe constant sur* $[a,b]$ *(convexité ou concavité)*

Alors

a) f *admet une racine unique dans* $[a,b]$.

b) la suite $(x_n)_n$ *converge vers* α, $\forall x_0 \in [a,b]$ *vérifiant* $f(x_0).f''(x_0) > 0$.

(la méthode de Newton converge vers l'unique solution α *de* $f(x) = 0$ *dans* $[a,b]$ *et ceci pour n'importe quel choix de* $x_0 \in [a,b]$*).*

De plus, cette convergence est quadratique tel que

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

c) Et on a l'estimation d'erreurs suivante :

$$|x_n - \alpha| \leq \frac{M}{2m} |x_n - x_{n-1}|^2$$

Où $M = \sup_{x\in[a,b]} |f''(x)|$, $\qquad m = \inf_{x\in[a,b]} |f'(x)|$.

Remarque 3.2.2 • *La méthode de bissection est inconditionnellement convergente, son inconvénient est sa lenteur pour obtenir la solution avec une grande précision. Elle peut servir pour démarrer d'autres méthodes plus performantes.*

• *La méthode des approximations successives est plus rapide que celle de bissection à condition qu'elle converge.*

• *La méthode de Newton-Raphson est la plus rapide, elle permet d'obtenir des solutions très précises en un nombre réduit d'itérations.*

Exemple 3.2.3 *Soit la fonction* $f(x) = x^3 + x - 1$

1) *Cette fonction admet une racine séparée sur* $\left[\frac{1}{2}, 1\right]$ *car:*

$$f\left(\frac{1}{2}\right).f(1) = (-0,375) \times 1 < 0 \quad et \quad f'(x) = 3x^2 + 1 > 0,$$

$$\exists! \alpha \in \left[\frac{1}{2}, 1\right], \quad f(\alpha) = 0$$

Vérifions maintenant les conditions du théorème de Newton :

La condition i) déjà vue dans 1)

ii) $f'(x) = 3x^2 + 1 \neq 0$, *sur* $\left[\frac{1}{2}, 1\right]$

iii) $f''(x) = 6x \neq 0$, *sur* $\left[\frac{1}{2}, 1\right]$

Le choix de x_0

$f''(x) > 0,\ f(1) > 0 \Longrightarrow f''(x).f(1) > 0 \Longrightarrow x_0 = 1$

$$M = \sup_{x \in \left[\frac{1}{2}, 1\right]} |f''(x)| = 6, \qquad m = \inf_{x \in \left[\frac{1}{2}, 1\right]} |f'(x)| = \frac{7}{4} \quad sur \quad \left[\frac{1}{2}, 1\right].$$

On calcule une approximation avec une précision $\varepsilon = 10^{-2}$.

$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n^3 + x_n - 1}{3x_n^2 + 1} = \frac{2x_n^3 + 1}{3x_n^2 + 1}$

Partant du choix de $x_0 = 1$, *on a* $x_1 = \frac{2x_0^3 + 1}{3x_0^2 + 1} = \frac{2(1)^3 + 1}{3(1)^2 + 1} = \frac{3}{4} = 0,75$

$$|x_1 - \alpha| \leq \frac{2M}{m} |x_1 - x_0|^2 = \frac{6}{2 \times \frac{7}{4}} |0,75 - 1|^2 \simeq 0,107 > \varepsilon$$

On calcule alors x_2

$x_2 = \frac{2x_1^3 + 1}{3x_1^2 + 1} = \frac{2\left(\frac{3}{4}\right)^3 + 1}{3\left(\frac{3}{4}\right)^2 + 1} \simeq 0,685$

$$|x_2 - \alpha| \leq \frac{M}{2m} |x_1 - x_0|^2 = \frac{2 \times 6}{\frac{7}{4}} |0,685 - 0,75|^2 \simeq 0,0017 \leq 10^{-2} = \varepsilon$$

d'ou x_2 *est la valeur approchée de à* 10^{-2}.

$$\alpha = 0,685 \overset{+}{-} 0,01$$

Remarque 3.2.4 *Nous pouvons remplacer la condition suivante*

$$f(x_0).f''(x_0) > 0.$$

par la suivante

$$\left|\frac{f(a)}{f'(a)}\right| \leq |b-a|\,, \left|\frac{f(b)}{f'(b)}\right| \leq |b-a|\,.$$

Exemple 3.2.5 *De notre exemple précédent* $f''(1).f(1) > 0$, *par la suivante*

$$\left|\frac{f(\frac{1}{2})}{f'(\frac{1}{2})}\right| = \left|\frac{-0,375}{1,75}\right| = 0,214 \leq \left|1-\frac{1}{2}\right|,\ \left|\frac{f(1)}{f'(1)}\right| = \left|\frac{1}{4}\right| \leq \left|1-\frac{1}{2}\right|.$$

Définition 3.2.6 *Une méthode itérative définie par* $x_{n+1} = g(x_n)$ *est d'ordre* p *ssi* $\exists k \in \mathbb{R}^+$ *telles que:*

$$\lim_{n\to+\infty} \frac{|x_{n+1}-\alpha|}{|x_n-\alpha|^p} = \lim_{n\to+\infty} \frac{|e_{n+1}|}{|e_n|^p} = k,$$

où le terme $e_n = x_n - \alpha$ *est l'erreur à l'étape* n.

Remarque 3.2.7 *Si* $p = 1$, *la convergence est dite linéaire.*

Si $p = 2$, *la convergence est dite quadratique.*

Si $p = 3$, *la convergence est dite cubique.*

Exemple 3.2.8 *la convergence la méthode de Newton est quadratique tel que*

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

$p = 2,\ \ k = \frac{f''(\alpha)}{2f'(\alpha)}$.

$$x_{n+1} - \alpha = x_n - \alpha + \frac{f(\alpha)-f(x_n)}{f'(x_n)} = x_n - \alpha + \alpha - x_n + \frac{(x_n-\alpha)^2}{2}\frac{f''(\zeta_n)}{f'(x_n)}$$

d'où

$$\frac{x_{n+1}-\alpha}{(x_n-\alpha)^2} = \frac{f''(\zeta_n)}{2f'(x_n)}.$$

alors

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

Ordre et rapidité de convergence

Supposons qu'on a deux méthodes telles que :

Méthode (1) : $\frac{|x_{n+1}-\alpha|}{|x_n-\alpha|} = 0.5$.

Méthode (2) : $\frac{|x_{n+1}-\alpha|}{|x_n-\alpha|^2} = 0.5$.

Méthode (1) :

$$|x_{n+1} - \alpha| = 0.5|x_n - \alpha| = (0.5)^2 |x_{n-1} - \alpha| = ... = (0.5)^n |x_0 - \alpha|.$$

Méthode (2) :

$$\begin{aligned} |x_{n+1} - \alpha| &= 0.5|x_n - \alpha|^2 = 0.5. \left((0.5) |x_{n-1} - \alpha|^2\right)^2 \\ &= (0.5)^3 |x_{n-1} - \alpha|^4 ... = (0.5)^{2^{n+1}-1} |x_0 - \alpha|^{2^{n+1}}. \end{aligned}$$

Question : Quel est pour chacune des deux méthodes, le nombre minimal d'itérations pour avoir une erreur$\leq 10^{-10}$, en supposant que $|x_0 - \alpha| = 0.5$?

méthode (1): $|x_{n+1} - \alpha| = (0.5)^n|x_0 - \alpha| \leq 10^{-10} \Rightarrow n \geq 33$.

méthode (2):$|x_{n+1} - \alpha| = (0.5)^{2^{n+1}-1} |x_0 - \alpha|^{2^{n+1}} \leq 10^{-4} \Rightarrow n \geq 5$.

Donc la deuxième méthode converge plus rapidement que la première.

D'où, plus l'ordre p est grand, plus vite l'erreur décroît.

La méthode de bissection (de dichotomie)

Cette méthode est utilisée pour calculer **les zéros** d'une fonction continue $f : \mathbb{R} \to \mathbb{R}$ dans un intervalle $[a, b]$.

L'algorithme

If $f(a) f(b) > 0$,

the method does (**not end with zéros**);

if $f(a) f(b) < 0$,

there is atleast one **zero** $\overline{\mathbf{x}}$ of f in the interval $[a, b]$;

as long as $((b - a) > \text{tol})$,

$x = \frac{a+b}{2}$;

iIf $f(x) = 0$,

$\overline{\mathbf{x}} = réro = x$ and method does;

if not,

if $f(a) f(x) < 0$

we pose $b = x, \quad f(b) = f(x)$;

if $f(x) f(b) < 0$

we pose $a = x, \quad f(a) = f(x)$;

and if

as long as

and if

Application

Ecrire un programme MATLAB qui implémente la méthode numérique de bissection et appliquez, cette méthode pour trouver **le zéro** de la fonction $f(x)$ dans l'intervalle $[1, 2]$ avec critère de convergence $\varepsilon = 0,001$;

$$f(x) = \ln(x) - x^2 + 2 = 0.$$

Utiliser le langage de programmation MATLAB pour exécuter l'algorithme de la résoluton.

clc

clear

a=1;

b=2;

ep=0,001;

f=@$(x)\log(x) - x.2 + 2$;

fplot(f,$[a, b]$)

grid on

$[x, N] = MyD$ichotomieFun(f, a, b, eps)

function$[x, N] = MyD$ichotomieFun(f, a, b, eps)

if $f(a) * f(b) > 0$

error (' Intervalle indapté');

end

N=0;

wile abs$(b - a)/2 > eps$

x=$(b - a)/2$

if $f(a) * f(x) > 0$

a=x;

elseif $f(a) * f(x) < 0$

b=x;

else

break

end

N=N+1;

end

end

x=

1,5645

N=

9

La méthode de Newton

Cette méthode numérique est la meilleur méthode utilisée pour calculer **les zéros** d'une fonction continue $f : \mathbb{R} \to \mathbb{R}$ dans un intervalle $[a, b]$ parce qu'elles est simple et rapide, le seule inconvénient est qu'elle utilise la fonction $f(x)$ et sa première dérivé $f'(x)$.

L'algorithme

If $f(a) f(b) > 0$,

The method does (**not end with zéros**);

if $f(a) f(b) < 0$,

There is atleast one **zero** $\overline{\mathbf{x}}$ of f in the interval $[a, b]$;

We take x_{est} as a first estimate for zero (of course in the meantime $[a, b]$)

We calculate $\Delta x = -\frac{f(x_{est})}{f'(x_{est})}$;

as long as (abs(Δx) >tol),

We define $x_{est} \to x_{est} + \Delta x$;

Calculate again $\Delta x = -\frac{f(x_{est})}{f'(x_{est})}$

as long as

$z\acute{e}ro = x_{est}$

end if

Application

Ecrire un programme MATLAB qui implémente la méthode numérique de Newton et appliquez cette méthode pour trouver **le zéro** de la fonction $f(x)$ dans l'intervalle

$[0,1;\ 0,5]$ avec critère de convergence $\varepsilon = 0,001$;

$$f(x) = \ln(x) - x^2 + 2 = 0.$$

Utiliser le langage de programmation MATLAB pour exécuter la l'algorithme de la résoluton.

clc

clear

a=0,1;

b=0,5;

ep=0,001;

f=@$(x)\log(x) - x.2 + 2$;

fplot(f,$[a, b]$)

grid on

$[x, N] = My$NewtonRaphsonFun(f, a, b, eps)

function$[x, N] =$ MyNewtonRaphsonFun(f, a, b, eps)

N=0;

$x = x_0$;

$x_1 = x\_f(x_0)/f\_prime(x_0)$

while abs $(x_1 - x) > eps$

$x_1 = x\_$

$x_1 = x\_f(x_0)/f\_prime(x_0)$;

N=N+1;

end

end

x=

1,1379

N=

7

Chapitre 4

Résolution numérique des équations différentielles ordinaires

Théorèmes d'existence et d'unicité

Soient $f : [t_0, T] \times \mathbb{R} \to \mathbb{R}$ une fonction suffisamment différentiable et $y_0 \in \mathbb{R}$, une fonction $x : [t_0, T] \to \mathbb{R}$ solution du problème de Cauchy:

$$\begin{cases} y'(t) = f(t, y(t)), \quad t \in [t_0, T] \\ y(t_0) = y_0 (\text{condition initiale }) \end{cases} \tag{4.1}$$

Théorème 4.0.9 *Soit $f : [t_0, T] \times \mathbb{R} \longrightarrow \mathbb{R}$ une fonction telle que:*

1. *f est continue sur $[t_0, T] \times \mathbb{R}$.*

2. *f est Lipschitzienne par rapport à la deuxième variable, c'est à dire il existe une constante $k > 0$ telle que pour tout $t \in [t_0, T]$ et y_1, $y_2 \in \mathbb{R}$, on ait:*

$$|f(t, y_2) - f(t, y_1)| \leq k|y_2 - y_1|.$$

Alors le problème (5.1) admet une unique solution $y \in C^1([t_0, T], \mathbb{R})$. On dit que le problème est bien posé.

Remarque 4.0.10 *Si f est définie et de classe C^1 dans $[t_0, T] \times \mathbb{R}$ et $\exists\ m \in \mathbb{R}^+$ tel que:*

$$\left|\frac{\partial f}{\partial y}\right| \leq m \quad \forall (t, y) \in [t_0, T] \times \mathbb{R}.$$

Alors, f satisfait la condition de Lipschitz par rapport à y sur $[t_0, T] \times \mathbb{R}$.

Exemple 4.0.11 *Considérons le problème de Cauchy suivant:*

$$\begin{cases} y'(t) = t\,|y(t)|\,, \quad t \in [1, 2], \\ y(1) = 1. \end{cases}$$

est continue dans $[1, 2] \times \mathbb{R}$ car : c'est la somme d'un polynôme. $f(t, y)$ est lipschitzienne par rapport à y $(k = 2)$ car:

$$|f(t, y_2) - f(t, y_1)| = |t\,|y_2| - t\,|y_1|\,| = t|\,|y_2| - |y_1|\,| \leq 2|y_2 - y_1| \quad \forall (y_1, y_2) \in \mathbb{R} \times \mathbb{R}.$$

D'après le théorème de Cauchy-Lipschitz il existe une seule solution y dans $[1, 2] \times \mathbb{R}$.

Exemple 4.0.12 *Considérons le problème de Cauchy suivant:*

$$y'(t) = -y^3(t) + e^{-\frac{t^2}{2}}$$

est continue dans $\mathbb{R} \times [1, 2]$ car: c'est la somme d'une exponentielle et d'un polynôme. $f(t, y)$ est lipschitzienne par rapport à y $(k = 12)$ car:

$$\frac{\partial f}{\partial y} = -3y^2 \Longrightarrow \left|\frac{\partial f}{\partial y}\right| = 3y^2 \leq 12.$$

D'après le théorème de Cauchy-Lipschitz il existe une seule solution y dans $\mathbb{R} \times [1, 2]$.

Position du problème

Résoudre une EDO explicitement n'est pas toujours facile sauf pour des cas simples.

Le principe de ces méthodes est de discrétiser l'intervalle $[a, b]$ en choisissant un pas de discrétisation h. Subdivisons l'intevalle $[0, T]$ comme suit : $0 = t_0 < t_1 < ... < t_n < t_{n+1} < ... < t_N = T$. pose: $t_n = t_0 + nh$, $h = t_{n+1} - t_n$.

1. Avec les outils numériques de résolution d'équations différentielles il n'est pas possible d'obtenir une solution pour toutes les valeurs de la variable t. On obtient plutôt une approximation de la solution analytique seulement pour certaines valeurs de t notées t_i et distancées d'une valeur $h_i = t_{i+1} - t_i$.

2. On note $y(t_i)$ la solution analytique de l'équation différentielle (1) en $t = t_i$, et y_i la solution approximative en $t = t_i$ à l'aide d'une méthode numérique.

4.1 Méthode d'Euler

Avant d'effectuer la première itération, il faut déterminer dans quelle direction on doit avancer à partir du point (t_0, y_0) pour obtenir le point (t_1, y_1), qui est une approximation du point $(t_1, y(t_1))$. L'équation différentielle (5.1) assure que:

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

$$y'(t_0) \approx \frac{y(t_1) - y(t_0)}{t_1 - t_0} = \frac{y(t_1) - y_0}{t_1 - t_0}$$

$$y(t_1) - y_0 = (t_1 - t_0)\, y'(t_0)$$

$$y(t_1) = y_0 + hf(t_0, y_0)$$

On remarque cependant que la pente de la solution analytique en $t = t_1$ est:

$$y'(t_1) = f(t_1, y(t_1)).$$

On ne connaît pas exactement $y(t_1)$, mais nous possédons l'approximation y_1 de $y(t_1)$. On doit alors utiliser l'expression. On pose: $y_n = y(t_n)$

$$\begin{aligned} y'(t_1) &\approx \frac{y(t_2) - y(t_1)}{t_2 - t_1} = \frac{y(t_2) - y_1}{t_2 - t_1} \\ y(t_2) - y_1 &= (t_2 - t_1)\, y'(t_1) \\ y(t_2) &= y_1 + hf(t_1, y(t_1)) \end{aligned}$$

On obtient la formule d'Euler:

$$\begin{cases} y_{n+1} = y_n + hf(t_n, y_n) \;\; t \in [0,T], \;\; t_{n+1} = t_n + h, \; 0 \leq n \leq N \\ y(0) = y_0 \text{ donné.} \end{cases}$$

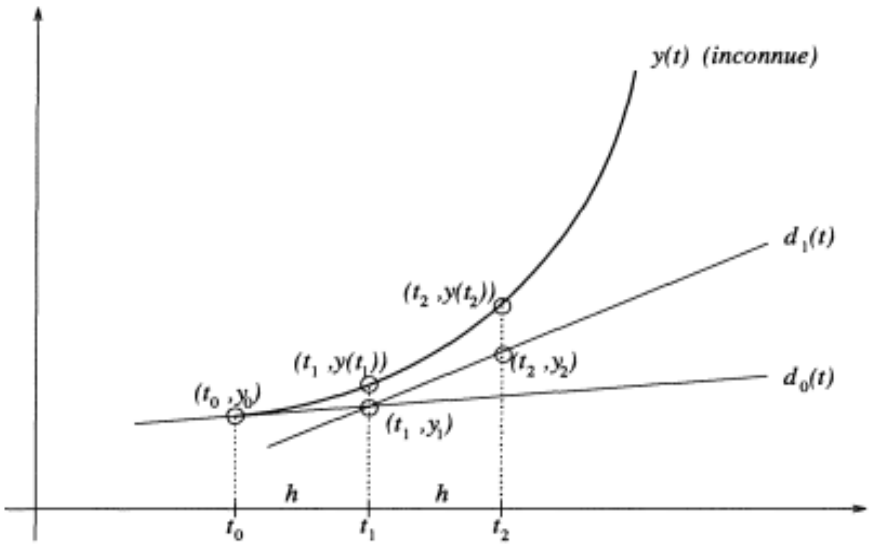


Figure 7.1: Méthode d'Euler

Exemple 4.1.1 *Soit le probléme de Cauchy*

$$\begin{cases} y'(t) = y(t) + t, \;\; t \in [0,1], \\ y(0) = 1. \end{cases} \tag{4.2}$$

On veut approcher, à 10^{-3}, *la solution de* (5.2) *en* $t = 1$ *à l'aide de la méthode d'Euler,*

en subdivisant l'intervalle $[0,1]$ *en dix parties égales, on calcule les valeurs du tableau:*

n	t_n	y_n
0	$0,0$	1
1	$0,1$	$1,1$
2	$0,2$	$1,22$
3	$0,3$	$1,362$
4	$0,4$	$1,5282$
5	$0,5$	$1,7210$
6	$0,6$	$1,9431$
7	$0,7$	$1,9431$
8	$0,8$	$1,4871$
9	$0,9$	$1,8158$
10	$1,0$	$1,18674$

On trouve $y(1) \simeq 3,187$. *La solution exacte de l'équation* (5.2) *est donnée par*

$$y(t) = 2e^t - t - 1,$$

ce qui donne $y(1) = 3,437$. *L'approximation calculée est donc trés grossiète.*

4.2 Méthode de Runge-Kutta

4.2.1 Méthodes de Runge Kutta d'ordre 2

Définition 4.2.1 *"Formule du rectangle ou point milieu " La formule du rectangle ou du point milieu est obtenue en remplaçant f par une constante égale à la valeur de f au milieu de* $[a,b]$

$$I(f)_{[a,b]} = \int_a^b f(x)dx = (b-a)f(\frac{a+b}{2}).$$

C'est le même principe que la méthode d'Euler. On cherche à approximer la solution exacte en utilisant la formule du point milieu pour l'approximation de l'intégrale $I(f)_{[a,b]}$ au lieu de la formule des rectangles à droite, on trouve:

$$\int_{t_n}^{t_{n+1}} y'(t)dt = \int_{t_n}^{t_{n+1}} f(t, y(t))dt = hf(t_n + \frac{h}{2}, y(t_n + \frac{h}{2})).$$

Le problème c'est qu'on n'a pas une valeur disponible $y(t_n + \frac{h}{2})$ pour calculer l'intégrale.

Une idée est de remplacer la valeur $y(t_n + \frac{h}{2})$ par la méthode d'Euler sur l'intervalle $[t_n, t_{n+1}]$ avec le pas $\widetilde{h} = \frac{h}{2}$. On trouve:

$$y(t_n + \widetilde{h}) = y(t_n) + \widetilde{h} f(t_n, y(t_n)).$$

En remplaçant $\widetilde{h}$ par $\frac{h}{2}$ on obtient:

$$y(t_n + \frac{h}{2}) = y(t_n) + \frac{h}{2} f(t_n, y(t_n)).$$

Donc:

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

Si on pose:

$$\begin{cases} K_1 = hf(t_n, y(t_n)),\ t_{n+1} = t_n + h, \\ K_2 = hf(t_n + \frac{h}{2}, y(t_n) + \frac{K_1}{2}),\ 0 \leq n \leq N. \end{cases}$$

On peut donner l'algorithme de Runge-Kutta d'ordre 2 comme:

$$y_{n+1} = y_n + K_2.$$

Exemple 4.2.2 *Soit l'équation différentielle déjà résolue par la méthode d'Euler*

$$\begin{cases} y'(t) = -y(t) + t + 1, \quad t \in [0, 1], \\ y(0) = 1. \end{cases}$$

On peut montrer que la solution analytique de cette équation est:

$$y(t) = e^{-t} + t,$$

On prend $h = 0,2$ *et* $f(t,y) = -y + t + 1$.

Itération 1*:*

$$\begin{cases} K_1 = hf(t_0, y_0) = (0,1) \times f(0;1) = (0,1) \times (0) = 0 \\ K_2 = hf(t_0 + \frac{h}{2}, y_0 + \frac{K_1}{2}) = (0,1) \times f(0,05;1) = (0,1) \times (0,05) = 0,005. \end{cases} ..$$

ce qui entraîne que $y_1 = y_0 + K_2 = 1 + 0,005 = 1,005$

Itération 2*:*

$$\begin{cases} K_1 = hf(t_1, y_1) = (0,1) \times f(0,1;1,005) = (0,1) \times (0,095) = 0,0095 \\ K_2 = hf(t_1 + \frac{h}{2}, y_1 + \frac{K_1}{2}) = (0,1) \times f(0,15;1,00975) = (0,1) \times (0,095) = 0,0095 \end{cases}$$

ce qui entraîne que: $y_2 = y_1 + K_2 = 1,005 + 0,0095 = 1,0145$.

Le tableau suivant rassemble les résultats des cinq premières itérations ce qui permet de comparer les solutions numérique et analytique et de constater la croissance de l'erreur, et la comparer avec les méthodes (Euler)

t_i	$y(t_i)$	y_i	$\lvert y(t_i) - y_i \rvert$
$0,0$	$1,0000000000$	$1,000$	$0,000000$
$0,2$	$1,0048374180$	$1,005$	$0,819 \times 10^{-7}$
$0,4$	$1,0187307798$	$1,0145$	$0,819 \times 10^{-7}$
$0,6$	$1,0408182207$	$1,0408184220$	$0,148 \times 10^{-6}$
$0,8$	$1,0703200460$	$1,0703202889$	$0,242 \times 10^{-6}$
1	$1,0703200460$	$1,0703202889$	$0,242 \times 10^{-6}$

4.2.2 Méthodes de Runge Kutta d'ordre 4

Au lieu d'utiliser la méthode du trapèze, On utilise la méthode de Simpson qui consiste à remplacer la fonction intégrée par une parabole passant par les points extrêmes et le point milieu. On a:

$$\int_a^b f(x)\,dx \simeq \frac{b-a}{6}\left(f(a)+4f\left(\frac{a+b}{2}\right)+f(b)\right),$$

Appliquée à l'intégrale $\int_{t_n}^{t_n+h} f(t,y(t))dt$, cela donne

$$\int_{t_n}^{t_{n+1}} f(t,y(t))dt \simeq \frac{h}{2}\left[f(t_n,y(t_n))+4f(t_{n+\frac{1}{2}},y(t_{n+\frac{1}{2}}))+f(t_{n+1},y(t_{n+1}))\right]$$

d'où la relation:

$$y_{n+1} = y_n + \frac{h}{2}\left[f(t_n,y(t_n))+4f(t_{n+\frac{1}{2}},y(t_{n+\frac{1}{2}}))+f(t_{n+1},y(t_{n+1}))\right]$$

Ici, on a une difficulté apparaît car l'équation présente deux inconnues: $y(t_{n+\frac{1}{2}})$ et $y(t_{n+1})$. Donc il faut estimer $4f(t_{n+\frac{1}{2}},y(t_{n+\frac{1}{2}}))$ et $f(t_{n+1},y(t_{n+1}))$ à partir de y_n, t_n et h.
On commençe par le terme $4f(t_{n+\frac{1}{2}},y(t_{n+\frac{1}{2}}))$: On le décompose en deux termes identiques

$$2f(t_{n+\frac{1}{2}},\underbrace{y(t_{n+\frac{1}{2}})}_{(a)}) + 2(t_{n+\frac{1}{2}},\underbrace{y(t_{n+\frac{1}{2}})}_{(b)})$$

$y^{(a)}_{n+\frac{1}{2}} = y_n + \frac{h}{2}f(t_n,y(t_n))$: Euler explicite
$y^{(b)}_{n+\frac{1}{2}} = y_n + \frac{h}{2}f(t_{n+\frac{1}{2}},y(t_{n+\frac{1}{2}}))$: Euler implicite
Donc on obtient:

$$y^{(b)}_{n+\frac{1}{2}} = y_n + \frac{h}{2}f(t_{n+\frac{1}{2}},\underbrace{y_n+\frac{h}{2}f(t_n,y(t_n))}_{\left(y^{(a)}_{n+\frac{1}{2}}\right)})$$

D'où:

$$y_{n+1} = y_n + \frac{h}{6}\left[\begin{array}{l} f(t_n, y_n) + 2f(t_{n+\frac{1}{2}}, y_n + \frac{h}{2}f(t_n, y_n)) \\ +2f(t_{n+\frac{1}{2}}, y_n + \frac{h}{2}f(t_{n+\frac{1}{2}}, y_n + \frac{h}{2}f(t_{n+\frac{1}{2}}, y_n + \frac{h}{2}f(t_n, y_n)) + f(t_{n+1}, y(t_{n+1}) \end{array}\right]$$

Puisque: $t_{n+\frac{1}{2}} = \frac{t_n + t_{n+1}}{2} = \frac{t_n + t_n + h}{2} = t_n + \frac{h}{2}$.

D'où la relation:

$$y_{n+1} = y_n + \frac{1}{6}\left[K_1 + 2K_2 + 2K_3 + f(t_{n+1}, y(t_{n+1})\right]$$

tel que

$$\begin{cases} K_1 = hf(t_n, y_n), \\ K_2 = hf(t_n + \frac{h}{2}, y_n + \frac{K_1}{2}) \\ K_3 = hf(t_n + \frac{h}{2}, y_n + \frac{K_2}{2}) \end{cases}$$

En estimation de $f(t_{n+1}, y(t_{n+1}))$, par la méthode du rectangle au milieu :

$$y_{n+1} \simeq y_n + hf(t_{n+\frac{1}{2}}, y_n + \frac{h}{2}f(t_{n+\frac{1}{2}}, y_{n+\frac{1}{2}})$$

Donc

$$y_{n+1} \simeq y_n + hf(t_{n+\frac{1}{2}}, y_n + \frac{h}{2}f(t_{n+\frac{1}{2}}, y_n + \frac{h}{2}f(t_n, y_n))$$

Finalement on obtient la relation explicite de RUNGE-KUTTA d'ordre 4:

$$y_{n+1} = y_n + \frac{1}{6}\left[K_1 + 2K_2 + 2K_3 + K_4\right],$$

avec

$$\begin{cases} K_1 = hf(t_n, y_n), \\ K_2 = hf(t_n + \frac{h}{2}, y_n + \frac{K_1}{2}) \\ K_3 = hf(t_n + \frac{h}{2}, y_n + \frac{K_2}{2}) \\ K_4 = hf(t_n + h, y_n + K_3) \end{cases}$$

Remarque 4.2.3 *Les méthodes de Runge Kutta d'ordre deux et quatre convergent vers*

la solution exacte plus vite que la méthode d'Euler.

Exemple 4.2.4 *Soit l'équation différentielle déjà résolue par la méthode d'Euler*

$$\begin{cases} y'(t) = y(t) + 1, \quad t \in [0,1], \\ y(0) = 0. \end{cases}$$

Calculer une valeur approchée de $y(1)$ *en utilisant le schéma de Runge-Kutta d'ordre* 4 *avec un pas* h *égal à* 0.1.
On montre immédiatement que cette solution est

$$y(t) = e^t - 1.$$

La méthode $(RK4)$ *donne sur cet exemple*

$$\begin{cases} K_1 = hf(t_n, y_n) = hy_n + h \\ K_2 = hf(t_n + \frac{h}{2}, y_n + \frac{K_1}{2}) = hy_n + \frac{h}{2}K_1 + h \\ K_3 = hf(t_n + \frac{h}{2}, y_n + \frac{K_2}{2}) = hy_n + \frac{h}{2}K_2 + h \\ K_4 = hf(t_n + h, y_n + K_3) = hy_n + hK_3 + h \end{cases}$$

$$y_{n+1} = y_n + \frac{1}{6}\left[K_1 + 2K_2 + 2K_3 + K_4\right]$$

t_i	$y(t_i)$	y_i	$\lvert y(t_i) - y_i \rvert$
$0,0$	$0,0000000$	$0,0000000$	$0,0000000$
$0,1$	$0,10517092$	$0,10517083$	$8,5 \times 10^{-8}$
$0,2$	$0,22140276$	$0,22140257$	$1,9 \times 10^{-7}$
$0,3$	$0,34985881$	$0,34985850$	$3,1 \times 10^{-7}$
$0,4$	$0,49182470$	$0,49182424$	$4,6 \times 10^{-7}$
$0,5$	$0,64872127$	$0,64872064$	$6,3 \times 10^{-7}$
$0,6$	$0,82211880$	$1,82211796$	$8,4 \times 10^{-7}$
$0,7$	$1,01375271$	$1,01375163$	$1,1 \times 10^{-6}$
$0,8$	$1,22551093$	$1,22553956$	$1,1 \times 10^{-6}$
$0,9$	$1,45960311$	$1,45960141$	$1,7 \times 10^{-6}$
$1,0$	$1,71828183$	$1,71827974$	$2,1 \times 10^{-6}$

Exemple 4.2.5 *Soit le problème de Cauchy:*

$$\begin{cases} y' = y - \frac{t}{y}, & t \in [0,1], \\ y(0) = 1. \end{cases}$$

La solution analytique est donnée par : $y = 2t + 1$.
L'application des différents processus a donné les résultats suivants sur l'intervalle $[0,1]$ *avec* $h = 0.2$ (*tous les résultats – même intermédiaires – doivent êtres calculés avec une précision de l'ordre de* 10^{-6})

t_i	y_i *Euler*	y_i $RK2$	y_i $RK4$	$y(t_i)$
$0,0$	$1,000000$	$1,000000$	$1,000000$	$1,000000$
$0,2$	$1,200000$	$1,186667$	$1,183229$	$1,183216$
$0,4$	$1,373333$	$1,348312$	$1,341667$	$1,341641$
$0,6$	$1,531495$	$1,493704$	$1,483281$	$1,483240$
$0,8$	$1,681085$	$1,627861$	$1,612514$	$1,612452$
1	$1,826949$	$1,754205$	$1,732142$	$1,732051$

Il est clair que la méthode de Runge Kutta d'ordre 4 *est la plus précise pour la résolution des équations différentielles.*

La méthode d'**Euler**

Lorsque le calcul analytique est difficile(ou impossible) le calcul numérique de l'intégral d'une fonction s'impose. L'une des manieres les plus naturelles pour calculer l'aire sous une courbe est d'approximer par une aire facilement calculable.

L'algorithme

$h = \frac{(b-a)}{n}$;

$t = z\acute{e}ros\ (1, n+1)$;

$y = z\acute{e}ros\ (1, n+1)$;

$t_1 = a$;

$y_1 = y_a$;

for $j = 1$ to n

$$y_{j+1} = y_j + h \times f(t_j, y_j)$$

$$t_{j+1} = a + h \times j$$

and if

soly= y;

solt= t;

Application

Soit l'équaton différentielle ordiniare suivante

$$y' = f(x, y)$$

avec la condition intiale:

$$y(x_0) = y_0$$

1−Trouver par la méthode de d'**Euler** la solution de cette équation différentielle dans l'intervalle donné $[x_0, x_n]$.

2− Utiliser le langage de programmation Matlab pour faire les calculs. $\begin{cases} y' = \frac{\cos(t)}{y} \\ y(0) = 2 \end{cases}$

$$
\begin{aligned}
& clc \\
& clear \\
x_0 &= 0,\ y_0 = 2 \\
x_n &= 10.14; \\
h &= 0,25 \\
f &= @(x,y)\cos(x)/y; \\
[x \text{ ; y-e; y-i,} y - \text{Si}] &= MyEulerFun(x_0, x_n, y_0, h, f); \\
& Plot\left(x, y-e,' rs', y-si,' r'_{+} LineWidth, 2\right); \\
Sol\_Exaxte &= sqrt(2 * (\sin(x) + 2)) \\
& hold\ on, \text{ plot(x,} Sol\_Exaxte,' K',' LineWidth', 2) \\
fonction(x \text{ , y-e, } y-i, y-si) &= MyEulerFun(x_0, x_n, y_0, h, f) \\
x &= x_0 : h : \ x_n; \\
n &= length(x); \\
y-e &= zeros(1,n) \\
ye(1) &= y_0; \\
y(i+1) &= y(i) + h * f(x(i), y-e(i)); \\
& end \\
y-i &= zeros(1,n) + y_0 \\
y\_i(1) &= y_0; \\
Erreur\_i &= 1 \\
while Erreur\_i &> = 0,001 \\
E\_i &= y\_i(n); \\
y\_i(i+1) &= y\_i(i) + h * f(x(i+1), y-i(i+1)) \\
& \text{end} \\
y - E = y; & \\
& \text{end}
\end{aligned}
$$

La méthode de Runge-Kutta d'ordre (4)

la méthode deRunge-Kutta d'ordre(4) est une amélioration de la méthode **d'Euler** en utilisant différentes modifiction et schémas de calcule. Pour résoudre (numériquement) l'équation différentielle:

$$y' = f(t, y(t)) \quad / \quad y(0) = y_0 \text{ avec } t_0 = 0 < ... < t_n.$$

On note $t_{n+1} = t_n + h$, on note par y_n une approximation de $y(t_n)$, on approche la solution de l'équation par formule:

$$\begin{cases} y_0 \\ y_{n+1} = y_n + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4) \end{cases}$$

$$\begin{cases} K_1 = hf(t_n, y_n), \\ K_2 = hf(t_n + \frac{h}{2}, y_n + \frac{K_1}{2}) \\ K_3 = hf(t_n + \frac{h}{2}, y_n + \frac{K_2}{2}) \\ K_4 = hf(t_n + h, y_n + K_3) \end{cases}$$

L'algorithme

$h = \frac{(b-a)}{n}$;

$t = zeros\ (1, n+1)$;

$y = zeros\ (1, n+1)$;

$t_1 = a : h : b$

$y_1 = y_a$;

for $j = 1$ to n

$$K_1 = hf(t_j, y_j),$$
$$K_2 = hf(t_j + \frac{h}{2}, y_j + \frac{K_1}{2})$$
$$K_3 = hf(t_j + \frac{h}{2}, y_j + \frac{K_2}{2})$$
$$K_4 = hf(t_j + h, y_j + K_3)$$

$$y_{j+1} = y_j + \frac{1}{6}\left(K_1 + 2K_2 + 2K_3 + K_4\right)$$

and if

soly= y;

solt= t;

Application

Soit l'équaton différentielle ordiniare suivante

$$y' = f(x, y) \quad \text{avec la condition intiale} \quad y\left(x_0\right) = y_0.$$

1−Trouver par la méthode de Rang-Kutta la solution de cette équation différentielle dans l'intervalle donée $[x_0, x_n]$.

2− Utiliser le langage de programmation Matlab pour faire les calculs. $\begin{cases} y' = \frac{\cos(t)}{y} \\ y\left(0\right) = 2 \end{cases}$

Programme principal

$$
\begin{aligned}
& clc \\
& clear \\
x_0 &= 0,\ y_0 = 2 \\
x_n &= 10.14; \\
h &= 0,25 \\
f &= @\left(x,y\right)\cos\left(x\right)/y; \\
[x\ ;\ \text{y-RK2; y-RK4}] &= MyRungKuttaFun(x_0,x_n,y_0,h,f); \\
& Plot\left(x,y-RK2,'rs',y-RK2,'b'_{+}LineWidth,2\right); \\
Sol\_Exaxte &= sqrt\left(2*\left(\sin\left(x\right)+2\right)\right) \\
& hold\ on,\ \text{plot(x,}Sol\_Exaxte,'K','LineWidth',2) \\
fonction\left(x\ ,\ \text{y-RK2, y-RK4}\right) &= MyRungKuttaFun(x_0,x_n,y_0,h,f) \\
x &= x_0 : h :\ x_n; \\
n &= length\left(x\right); \\
y &= zeros\left(1,n\right) \\
y\left(1\right) &= y_0; \\
for\ \ i = 1 : n-1 & \\
x-milieu &= x\left(i\right)+h/2; \\
x-milieu &= x\left(i\right)+h/2*f\left(x-milieu,y-milieu\right); \\
& end \\
y-RK2 &= y, \\
fot\ \ i = 1 : n-1 & \\
K_1 &= h*f(x\left(i\right),y\left(i\right)) \\
K_2 &= h*f(x\left(i\right)+\frac{h}{2},y\left(i\right)+\frac{K_1}{2}) \\
K_3 &= h*f(x\left(i\right)+\frac{h}{2},y\left(i\right)+\frac{K_2}{2}) \\
K_4 &= h*f(x\left(i\right)+h,y\left(i\right)+K_3) \\
y\left(i+1\right) &= y\left(i\right)+\frac{1}{6}*\left(K_1+2*K_2+2*K_3+K_4\right); \\
& \text{end} \\
y-RK-4 &= y;
\end{aligned}
$$

Chapitre 5

Résolution numérique des systèmes d'équations linéaires

Position du problème

On appelle système linéaire d'ordre n, $(n \in \mathbb{N}^*)$, une expression de la forme

$$AX = b \tag{5.1}$$

Où $A = (a_{i,j}), 1 \leq i,\ j \leq n$, désigne une matrice carrée d'ordre n de nombres réels ou complexes, $b = (b_i), 1 \leq i \leq n$, un vecteur colonne réel ou complexe et $X = (x_i)$, $1 \leq i \leq n$, est le vecteur des inconnues du système. La relation (1) équivaut aux équations

$$\sum_{j=1}^{n} a_{ij}x_j = b_i,\ \ i = 1, ..., n.$$

La matrice A est dite régulière (inversible) si $det(A) \neq 0$, on a existence et unicité de la solution X si et seulement si la matrice A est inversible.
On cherche à résoudre le système linéaire (1)

a) Méthodes directes: Ce sont des méthodes qui permettent d'obtenir la solution X de (1), si l'ordinateur faisait des calculs exacts, en un nombre fini (en relation avec n) d'opérations élémentaires.

Exemple 5.0.6 *Soit système linéaire $AX = b$.*

$$\begin{pmatrix} 2 & 3 & -1 \\ 0 & -2 & -1 \\ 0 & 0 & -5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 5 \\ -7 \\ -15 \end{pmatrix}, \begin{cases} 2x_1 + 3x_2 - 1x_3 = 5 \\ -2x_2 - x_3 = -7 \\ -5x_3 = -15 \end{cases} \Longleftrightarrow \begin{cases} x_3 = \frac{-15}{-5} = 3 \\ x_2 = 2 \\ x_1 = 1 \end{cases}$$

5.1 Méthode d'élimination de Gauss

Principe de la méthode : Déterminer une matrice M inversible telle que la matrice MA soit triangulaire supérieure. Alors

$$AX = b \Longleftrightarrow (MA)\, X = Mb$$

ensuite résoudre le système triangulaire supérieur $(MA)\, X = Mb$ par l'algorithme de remontée. Autrement dit

$$(A, b) \overset{\text{transformation}}{\longrightarrow} (A^{(n)}, b^{(n)}),$$

où $A^{(n)}$ est une matrice triangulaire supérieure, puis on résout le système triangulaire.

$$A^{(n)} X = b^{(n)}$$

Algorithme d'élimination de Gauss:

$$A = \begin{pmatrix} a_{11} & a_{12} & . & . & a_{1n} \\ a_{21} & a_{22} & . & & a_{2n} \\ . & . & . & & . \\ . & & & & . \\ a_{n1} & . & . & a_{11} & a_{nn} \end{pmatrix}, \; X = \begin{pmatrix} x_1 \\ x_2 \\ . \\ . \\ x_n \end{pmatrix}, b = \begin{pmatrix} b_1 \\ b_2 \\ . \\ . \\ b_n \end{pmatrix}$$

Soit (S) le système de départ:

$$\left\{\begin{array}{c} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + +...a_{2n}x_n = b_2 \\ : \\ a_{1n}x_1 + a_{n2}x_2 + ... + a_{nn}x_n = b_n \end{array}\right. \qquad \text{(S)}$$

On considère le système 3×3 suivant dont on suppose qu'il admet une et une seule solution

$$\left\{\begin{array}{l} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \end{array}\right.$$

$$(A/b)^{(0)} \quad \left(\begin{array}{ccccc} a_{11} & a_{12} & a_{13} & . & b_1 \\ a_{21} & a_{22} & a_{23} & . & b_2 \\ a_{31} & a_{32} & a_{33} & . & b_3 \end{array}\right)$$

On rappelle que les systèmes $(A/b)^{(1)}$

$$(A/b)^{(1)} \quad \left(\begin{array}{ccccc} a_{11} & a_{12} & a_{13} & . & b_1 \\ 0 & \left(a_{22} - \frac{a_{21}}{a_{11}}a_{12}\right) & \left(a_{23} - \frac{a_{21}}{a_{11}}a_{13}\right) & . & b_2 - \frac{a_{21}}{a_{11}}b_1 \\ 0 & \left(a_{32} - \frac{a_{31}}{a_{11}}a_{12}\right) & \left(a_{33} - \frac{a_{31}}{a_{11}}a_{13}\right) & . & b_3 - \frac{a_{31}}{a_{11}}b_1 \end{array}\right)$$

abrégé en

$$(A/b)^{(1)} \quad \left(\begin{array}{ccccc} a_{11} & a_{12} & a_{13} & . & b_1 \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & . & b_2^{(1)} \\ 0 & a_{32}^{(1)} & a_{33}^{(1)} & . & b_3^{(1)} \end{array}\right)$$

On rappelle que les systèmes $(A/b)^{(2)}$

$$(A/b)^{(2)} \quad \begin{pmatrix} a_{11} & a_{12} & a_{13} & . & b_1 \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & . & b_2^{(1)} \\ 0 & 0 & \left(a_{33}^{(1)} - \frac{a_{32}}{a_{22}} a_{23}^{(1)}\right) & . & b_3 - \frac{a_{32}}{a_{22}} b_2^{(1)} \end{pmatrix}$$

abrégé en

$$(A/b)^{(2)} \quad \begin{pmatrix} a_{11} & a_{12} & a_{13} & . & b_1 \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & . & b_2^{(1)} \\ 0 & 0 & a_{33}^{(2)} & . & b_3^{(2)} \end{pmatrix}$$

Exemple 5.1.1 *Soit système linéaire* $AX = b$.

$$A = \begin{pmatrix} 2 & 3 & -1 \\ 4 & 4 & -3 \\ -2 & 3 & -2 \end{pmatrix} \quad b = \begin{pmatrix} 5 \\ 3 \\ -2 \end{pmatrix}$$

$$(A/b)^{(0)} \; : \; \begin{pmatrix} 2 & 3 & -1 & . & 5 \\ 4 & 4 & -3 & . & 3 \\ -2 & 3 & -2 & . & -2 \end{pmatrix}, \; (A/b)^{(1)} : \begin{pmatrix} 2 & 3 & -1 & . & 5 \\ 0 & -2 & -1 & . & -7 \\ 0 & 6 & -3 & . & 3 \end{pmatrix},$$

$$(A/b)^{(2)} \; : \; \begin{pmatrix} 2 & 3 & -1 & . & 5 \\ 0 & -2 & -1 & . & -7 \\ 0 & 0 & -6 & . & -18 \end{pmatrix}$$

Que l'on peut écrire sous la forme:

$$\begin{cases} 2x_1 + 3x_2 - x_3 = 5 \\ -2x_2 - x_3 = -7 \\ -6x_3 = -18 \end{cases} \quad \begin{cases} x_3 = \frac{-18}{-6} = 3 \\ x_2 = \frac{-1}{2}(x_3 - 7) = \frac{-1}{2}(3 - 7) = 2 \\ x_1 = \frac{1}{2}(5 - 3x_2 + x_3) = \frac{1}{2}(5 - 3 \times 2 + 3) = 1 \end{cases}$$

$detA = \prod_{i=1}^{n} a_{ii}^{i}$, $detA = (2) \times (-2) \times (-6) = 24$.

Exemple 5.1.2 *:Soit système linéaire.*

$$\begin{cases} x_1 + x_2 + 3x_4 = 4 \\ 2x_1 + x_2 - x_3 + x_4 = 1 \\ 3x_1 - x_2 - x_3 + 2x_4 = 1 \\ -x_1 + 2x_2 + 3x_3 - x_4 = 1 \end{cases}$$

$$AX = b \quad A = \begin{pmatrix} 1 & 1 & 0 & 3 \\ 2 & 1 & -1 & 1 \\ 3 & -1 & -1 & 2 \\ -1 & 2 & 3 & -1 \end{pmatrix} \quad b = \begin{pmatrix} 4 \\ 1 \\ -3 \\ 4 \end{pmatrix} \quad (A/b)^{(0)} : \left(\begin{array}{cccc.c} 1 & 1 & 0 & 3 & . & 4 \\ 2 & 1 & -1 & 1 & . & 1 \\ 3 & -1 & -1 & 2 & . & -3 \\ -1 & 2 & 3 & -1 & . & 4 \end{array}\right),$$

$$(A/b)^{(1)} : \left(\begin{array}{cccc.c} 1 & 1 & 0 & 3 & . & 4 \\ 0 & -1 & -1 & -5 & . & 1 \\ 0 & -4 & -1 & -7 & . & -15 \\ 0 & 3 & 3 & 2 & . & 8 \end{array}\right), \quad (A/b)^{(2)} : \left(\begin{array}{cccc.c} 1 & 1 & 0 & 3 & . & 4 \\ 0 & -1 & -1 & -5 & . & -7 \\ 0 & 0 & 3 & 13 & . & 13 \\ 0 & 0 & 0 & -13 & . & -13 \end{array}\right)$$

Que l'on peut écrire sous la forme:

$$\begin{cases} x_1 + x_2 + 3x_4 = 4 \\ -x_2 - x_3 - 5x_4 = -7 \\ 3x_3 + 13x_4 = 13 \\ -13x_4 = -13 \end{cases}$$

Nous obtenous $X^t = (-1, 2, 0, 1)$, $\quad detA = \prod_{i=1}^{n} a_{ii}^{i}$, $detA = (1) \times (-1) \times (3) \times (-13) = 39$.

b) Méthodes itératives : Ce sont des méthodes qui consistent à construire une suite de vecteurs $X^{(k)}$ convergeant vers la solution X.

Définition 5.1.3 *La méthode itérative* (2) *est converge si*

$$\underset{k\to+\infty}{X^{(k)}} = \overline{X}, \qquad \forall X^{(0)} \in \mathbb{R}^n.$$

Principe des méthodes itératives:

Ecrivons d'abord la matrice A sous la forme $A = M - N$ où M est inversible, alors

$$AX = b \Longleftrightarrow (M - N)\, X = b \Longleftrightarrow MX = NX + b$$

Multiplions les deux côtés par M^{-1}, on aura

$$\begin{aligned} X &= M^{-1}NX + M^{-1}b \qquad (5.2) \\ &\begin{cases} X^{(0)} \text{ donné} \\ X^{(k+1)} = M^{-1}NX^{(k)} + M^{-1}b = BX^{(k)} + C, \quad B = M^{-1}N, \quad C = M^{-1}b \end{cases} \end{aligned}$$

Le principe de toutes les méthodes itératives est le suivant :

- choisir un vecteur $X^{(0)} \in \mathbb{R}^n$,
- générer la suite $(X^{(k)})_{k\in N}$ telle que $X^{(k+1)} = M^{-1}NX^{(k)} + M^{-1}b$,
- si la suite $(X^{(k)})_{k\in N}$ converge vers X^* , alors X^* est la solution du système $AX = b$.

Principales méthodes itératives

On considère la décomposition suivante de la matrice A

$$A = D - E - F = \begin{pmatrix} & & -F \\ & D & \\ -E & & \end{pmatrix}$$

avec:

D la diagonale de A

$-E$ la partie inférieure stricte

$-F$ la partie supérieure stricte.

Exemple 5.1.4 *Soit* $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$ *alors*

$$D = \begin{pmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{pmatrix}, \; E = \begin{pmatrix} 0 & 0 & 0 \\ -a_{21} & 0 & 0 \\ -a_{31} & -a_{32} & 0 \end{pmatrix}, \; F = \begin{pmatrix} 0 & -a_{12} & -a_{13} \\ 0 & 0 & -a_{23} \\ 0 & 0 & 0 \end{pmatrix}$$

On suppose que $\forall i = 1, 2, ..., n, \;\; a_{ii} \neq 0_{\mathbb{R}}$ (on peut s'y ramener si A est inversible).

i) **Méthode de Jacobi :** $M = D, \; N = E + F$

$$\begin{cases} X^{(0)} \in \mathbb{R}^n \;\; \textbf{donné} \\ X^{(k+1)} = D^{-1}(E+F)X^{(k)} + D^{-1}b = B_J X^{(k)} + C, \quad k \geq 0 \end{cases}$$

$B_J = D^{-1}(E+F)$ est appelée la matrice de Jacobi.

A chaque étape, on calculer les N composantes $x_1^{(k+1)}, ..., x_n^{(k+1)}$ du vecteur $X^{(k+1)}$.

$$\begin{cases} x_1^{(k+1)} = \frac{1}{a_{11}} \left[b_1 - a_{12}x_1^{(k)} - ... - a_{1n}x_n^{(k)} \right] \\ x_2^{(k+1)} = \frac{1}{a_{22}} \left[b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)} ... - a_{2n}x_n^{(k)} \right] \\ \qquad\qquad . \\ \qquad\qquad . \\ x_n^{(k+1)} = \frac{1}{a_{nn}} \left[b_n - a_{n1}x_1^{(k)} - ... - a_{nn-1}x_{n-1}^{(k)} \right] \end{cases}$$

Exemple 5.1.5 *Soit système linéaire* $AX = b$.

$$A = \begin{pmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{pmatrix} \quad b = \begin{pmatrix} 2 \\ 4 \\ 10 \end{pmatrix}$$

$$\left\{\begin{array}{c} 4x_1 - x_2 = 2 \\ -2x_1 + 4x_2 - x_3 = 4 \\ -x_2 + 4x_3 = 10 \end{array}\right. \iff \left\{\begin{array}{l} x_1 = \frac{1}{4}(x_2 + 2) \\ x_2 = \frac{1}{4}(2x_1 + x_3 + 4) \\ x_3 = \frac{1}{4}(x_2 + 10) \end{array}\right.$$

$$\iff \left\{\begin{array}{l} X^{(0)} = (0,0,0)^T \\ x_1^{(k+1)} = \frac{1}{4}\left(x_2^{(k)} + 2\right) \\ x_2^{(k+1)} = \frac{1}{4}\left(2x_1^{(k)} + x_3^{(k)} + 4\right) \\ x_3^{(k+1)} = \frac{1}{4}\left(x_2^{(k)} + 10\right) \end{array}\right.$$

k	x_1	x_2	x_3
0	0	0	0
1	0,5	1	2,5
2	0,75	1,75	2,75
3	0,9375	1,8750	2,9375

5.2 Méthode de Gauss-Seidel

ii) **Méthode de Gauss-Seidel :** $M = D - E,\ N = F$ On pourrait améliorer la méthode en utilisant les quantités déja calculées, on calcule successivent les N composantes

$$\left\{\begin{array}{c} X^{(0)} \in \mathbb{R}^n \text{ donné} \\ X^{(k+1)} = (D-E)^{-1} F X^{(k)} + (D-E)^{-1} b = B_{GS} X^{(k)} + C, \end{array}\right.$$

$B_{GS} = (D-E)^{-1} F$ est appelée la matrice de Gauss-Seidel.

A chaque étape, on calcule les N composantes $x_1^{(k)}, ..., x_n^{(k+1)}$ du vecteur $X^{(k+1)}$.

$$\begin{cases} x_1^{(k+1)} = \frac{1}{a_{11}} \left[b_1 - a_{12}x_2^{(k)} - ... - a_{1n}x_n^{(k)} \right] \\ x_2^{(k+1)} = \frac{1}{a_{22}} \left[b_2 - a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)} ... - a_{2n}x_n^{(k)} \right] \\ \qquad\qquad\qquad\qquad . \\ \qquad\qquad\qquad\qquad . \\ x_n^{(k+1)} = \frac{1}{a_{nn}} \left[b_n - a_{n1}x_1^{(k+1)} - ... - a_{nn-1}x_{n-1}^{(k+1)} \right] \end{cases}$$

Elle est inversible si $a_{ii} \neq 0$, $1 \leq i \leq n$.

Exemple 5.2.1 *Soit système linéaire* $AX = b$.

$$A = \begin{pmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{pmatrix} \quad b = \begin{pmatrix} 2 \\ 4 \\ 10 \end{pmatrix}$$

$$\begin{cases} 4x_1 - x_2 = 2 \\ -2x_1 + 4x_2 - x_3 = 4 \\ -x_2 + 4x_3 = 10 \end{cases} \iff \begin{cases} X^{(0)} = (0,0,0)^T \\ x_1^{(k+1)} = \frac{1}{4}\left(x_2^{(k)} + 2 \right) \\ x_2^{(k+1)} = \frac{1}{4}\left(2x_1^{(k+1)} + x_3^{(k)} + 4 \right) \\ x_3^{(k+1)} = \frac{1}{4}\left(x_2^{(k+1)} + 10 \right) \end{cases}$$

k	x_1	x_2	x_3
0	0	0	0
1	0,5	0,1250	2,7812
2	0,7818	0,8906	2,9726
3	0,9726	1,9863	2,9965

Condition nécessaire et suffisante de convergence

Soit P le polynôme caractéristique de A.

λ valeur propre de $A \iff P(\lambda) = det(A - \lambda I) = 0$.

c'est à dire λ est une racine du polynôme caractéristique P.

Soit

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

Les valeurs propres de la matrice B sont $\lambda_1 = 3$, $\lambda_2 = 2$, $\lambda_3 = 1$.

Thèréme de la convergence

Ce sont des méthodes qui consistent à construire une suite de vecteurs $X^{(k)}$ convergeant vers la solution $\overline{X}$.

Définition 5.2.2 *La méthode itérative* (2) *est convergente si*

$$\underset{k\to+\infty}{X^{(k)}} = \overline{X}, \qquad \forall X^{(0)} \in \mathbb{R}^n.$$

Théorème 5.2.3 *La suite* $(X^{(k)})_{k\in N}$ *définie par* $\forall X^{(0)} \in \mathbb{R}^n$ *quelconque*

$$X^{(k+1)} = BX^{(k)} + M^{-1}b,$$

Une condition nécessaire et suffisant de convergence de la méthode itérative (S) *est que le* $\rho(B) < 1$

ρ : *le rayon specterale de* B $(\rho(B) = \max\limits_{1\le i\le n} |\lambda_i| < 1)$

λ_i : *Valeur propre de* B.

et une condition suffisant $\|B\| < 1$ *car* $\rho(B) \le \|B\|$.

le nombre d'itérations

Critère d'arrêt: soit $k \in \mathbb{N}$. On a

$$\frac{\|B\|^k}{1-\|B\|} \left\|X^{(1)} - X^{(0)}\right\| \le \varepsilon$$

C'est-à-dire pour avoir $\left\|X^{(k)} - \overline{X}\right\| \leq \varepsilon$, il suffisant de prendre:

$$\frac{\|B\|^k}{1-\|B\|} \left\|X^{(1)} - X^{(0)}\right\| \leq \varepsilon.$$

Exemple 5.2.4 *Soit système linéaire $AX = b$.*

$$A = \begin{pmatrix} 2 & -1 & 1 \\ 2 & 2 & 2 \\ -1 & -1 & 2 \end{pmatrix}$$

Etudier la convergence des méthodes de Jacobi et Gauss-Seidel appliquées à système, pour tout choix de $X^{(0)} \in \mathbb{R}^3$, en utilisant le rayon spectral des matrices d'itérations. Conclure.

Estimer le nombre d'itérations nécessaires à l'approximation $X^{(0)} = (0,0,0)^T$ de la solution de ce système à $0,001$ près par la méthode de Gauss-Seidel.

Réponse: On trouve les résultats suivants:

$$B_J = \begin{pmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ -1 & 0 & -1 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix}, \quad B_{GS} = \begin{pmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & -\frac{1}{2} \end{pmatrix}$$

avec $\rho(B_{J2}) = 1,118 > 1$, $\rho(\ B_{GS2}) = 0,5 < 1$, d'où la méthode de Gauss-Seidel converge $\forall X^{(0)} \in \mathbb{R}^3$, alors que celle de Jacobi ne converge pas $\forall X^{(0)} \in \mathbb{R}^3$.

$$\frac{\|B\|^k}{1-\|B\|} \left\|X^{(1)} - X^{(0)}\right\| \leq \varepsilon$$

$$\frac{(0,5)^k}{1-0,5} \left\|X^{(1)} - (0,0,0)\right\| \leq 0,001$$

Exemple 5.2.5 *Soient les systèmes linéaires* $A_iX = b_i,\ \ i = 1, 2, 3.$

$$A_1 = \begin{pmatrix} 1 & 2 & -2 \\ 1 & 1 & 1 \\ 2 & 2 & 1 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 2 & -1 & 1 \\ 2 & 2 & 2 \\ -1 & -1 & 2 \end{pmatrix}, \quad A_3 = \begin{pmatrix} 1 & 4 & 4 \\ 2 & -9 & 0 \\ 0 & -8 & -6 \end{pmatrix},$$

Etudier la convergence des méthodes de Jacobi et Gauss-Seidel appliquées à chaque système, pour tout choix de $X^{(0)} \in \mathbb{R}^3$ *, en utilisant le rayon spectral des matrices d'itérations. Conclure.*

Réponse On trouve les résultats suivants:

$$B_{J1} = \begin{pmatrix} 0 & -2 & 2 \\ -1 & 0 & -1 \\ -2 & -2 & 0 \end{pmatrix}, \quad B_{GS1} = \begin{pmatrix} 0 & -2 & 2 \\ 0 & 2 & -3 \\ 0 & 0 & 2 \end{pmatrix}$$

avec $\rho(B_{J1}) = 0 < 1, \rho(\ B_{GS1}) = 2 > 1$, d'où la méthode de Jacobi converge $\forall X^{(0)} \in \mathbb{R}^3$, alors que celle de Gauss-Seidel ne converge pas $\forall X^{(0)} \in \mathbb{R}^3$.

$$B_{J2} = \begin{pmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ -1 & 0 & -1 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix}, \qquad B_{GS2} = \begin{pmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & -\frac{1}{2} \end{pmatrix}$$

avec $\rho(B_{J2}) = 1,118 > 1$, $\rho(\ B_{GS2}) = 0,5 < 1$, d'où la méthode de Gauss-Seidel converge $\forall X^{(0)} \in \mathbb{R}^3$, alors que celle de Jacobi ne converge pas $\forall X^{(0)} \in \mathbb{R}^3$.

$$B_{J3} = \begin{pmatrix} 0 & -\frac{1}{4} & -\frac{1}{4} \\ \frac{2}{9} & 0 & 0 \\ 0 & -\frac{4}{3} & 0 \end{pmatrix}, \qquad B_{GS3} = \begin{pmatrix} 0 & -\frac{1}{4} & -\frac{1}{4} \\ 0 & \frac{1}{8} & \frac{1}{8} \\ 0 & -\frac{1}{6} & -\frac{1}{6} \end{pmatrix}$$

avec $\rho(B_{J3}) = 0,44 < 1, \rho(\ B_{GS3}) = 0,018 < 1$, d'où les deux méthodes convergent $\forall X^{(0)} \in \mathbb{R}^3$, mais comme $\rho(B_{GS3}) < \rho(B_{J3})$ alors, la méthode qui converge plus rapide-

ment est celle de Gauss-Seidel.

Convergence des méthodes itératives

Définition 5.2.6 *Soit $A = (a_{ij})_{i;j=1}^{n}$ est une matrice carrée d'ordre n; alors on peut définir les normes matricielles suivantes*

$$\begin{aligned} \|A\|_1 &= \max_j \sum_{i=1}^{n} |a_{ij}|, \\ \|A\|_\infty &= \max_i \sum_{j=1}^{n} |a_{ij}|, \\ \|A\|_2 &= \left(\sum_{i=1}^{n} \sum_{j=1}^{n} |a_{ij}|^2 \right)^{\frac{1}{2}}. \end{aligned}$$

Cas

$$\begin{aligned} \|X\|_1 &= \sum_{i=1}^{n} |x_i|, \\ \|X\|_\infty &= \max \left(|x_1|, |x_2|, ..., |x_n| \right), \\ \|X\|_2 &= \sqrt{x_1^2 + x_2^2 + ... + x_n^2}. \end{aligned}$$

Théorème 5.2.7 *Si pour une norme matricielle donnée on a $\|B\| < 1$ alors la méthode itérative* (5.1) *est convergente.*

Définition 5.2.8 *Soit $A = (a_{ij})_{i;j=1}^{n}$ est une matrice carée d'ordre m. On dit que A est une matrice à diagonale strictement dominante si*

$$|a_{ii}| > \sum_{j=1 \ \ j \neq i}^{n} |a_{ij}| \qquad or \qquad |a_{jj}| > \sum_{i=1 \ \ i \neq j}^{n} |a_{ij}|.$$

Théorème 5.2.9 *Si la matrice A du système* (4.2) *est à diagonale strictement dominante alors toute méthode itérative de résolution est convergente.*

Exemple 5.2.10 *On considère le système* (S) *défini par:*

$$\begin{cases} 4x_1 - x_2 + x_3 = 6, \\ x_1 + 7x_2 - 2x_3 = -8, \\ 2x_1 - x_2 - 5x_3 = 8. \end{cases} \quad (1)$$

Calcul B_J, $\|B_J\|_1$, $\|B_J\|_\infty$, $\|B_J\|_2$.

$$A = \begin{pmatrix} 4 & -1 & 1 \\ 1 & 7 & -2 \\ 2 & -1 & -5 \end{pmatrix}$$

$detA = 126 \neq 0$

$$D = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & -5 \end{pmatrix}, \quad detD = 140 \neq 0, \quad D^{-1} = \begin{pmatrix} \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{7} & 0 \\ 0 & 0 & -\frac{1}{5} \end{pmatrix}$$

$$E = \begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ -2 & 1 & 0 \end{pmatrix} F = \begin{pmatrix} 0 & 1 & -1 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{pmatrix}$$

$$B_J = D^{-1}(E+F) = \begin{pmatrix} \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{7} & 0 \\ 0 & 0 & -\frac{1}{5} \end{pmatrix} \begin{pmatrix} 0 & 1 & -1 \\ -1 & 0 & 2 \\ -2 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{7} & 0 & 2 \\ -\frac{2}{5} & \frac{1}{5} & 0 \end{pmatrix}$$

La matrice A *est diagonal fortement dominante :*

$$\begin{aligned} |a_{11}| &= 4 > |-1| + |1| = |a_{12}| + |a_{13}| \\ |a_{22}| &= 7 > |1| + |-2| = |a_{21}| + |a_{23}| \\ |a_{33}| &= 5 > |2| + |-1| = |a_{31}| + |a_{32}| \end{aligned}$$

$$
\begin{aligned}
\|B_J\|_1 &= \max_j \sum_{i=1}^{3} |a_{ij}| = \max\left(\left|-\frac{1}{7}\right| + \left|-\frac{2}{5}\right|, \left|\frac{1}{4}\right| + \left|\frac{1}{5}\right|, \left|-\frac{1}{4}\right| + \left|\frac{2}{7}\right|\right) \\
&= \max\left(\frac{19}{35}, \frac{9}{20}, \frac{15}{28}\right) = \frac{19}{35} < 1 \\
\|B_J\|_\infty &= \max_i \sum_{j=1}^{3} |a_{ij}| = \max\left(\left|\frac{1}{4}\right| + \left|-\frac{1}{4}\right|, \left|-\frac{1}{7}\right| + \left|\frac{2}{7}\right|, \left|-\frac{2}{5}\right| + \left|\frac{1}{5}\right|\right) = \max\left(\frac{1}{2}, \frac{3}{7}, \frac{3}{5}\right) = \frac{3}{5} < 1 \\
\|B_J\|_2 &= \left(\sum_{j=1}^{3} |a_{ij}|^2\right)^{\frac{1}{2}} = \left(\left(\frac{1}{4}\right)^2 + \left(-\frac{1}{4}\right)^2 + \left(-\frac{1}{7}\right)^2 + \left(-\frac{2}{5}\right)^2 + \left(\frac{1}{5}\right)^2\right)^{\frac{1}{2}} = 0,6534 < 1
\end{aligned}
$$

Parce que $\|B_J\|_1 < 1$, $\|B_J\|_\infty < 1$, $\|B_J\|_2 < 1$ *donc la méthode de Jacobi est convergente.*

La méthode de gauss

L'algorithme

For $k = 1$ to $n - 1$

for $i = k + 1$ to n

$m = a(i,k)/a(k,k)$

for $j = k + 1$ to n

$a(i,j) = a(i,k) - m \times a(k,j)$

end if

$b(i) = b(i) - m \times b(k)$

end if

end if

$x(n) = b(n)/a(n,n)$;

for $j = i + 1$ to n

$s = s - a(i,j)^* x(j)$

end if

$x(i) = s/a(i,i)$;

end if

Application

Ecrire un programme MATLAB qui implémente la méthode numérique de résolution numérique des systèmes d'équations linéaires (la méthode de gauss) et appliquer cette méthode pour trouver la solution des systèmes d'équations linéaires suivants:

$$\begin{cases} 10x_1 - 5x_2 = 15 \\ -5x_1 + 12x_2 + 2x_3 = 0 \\ 2x_2 + 8x_3 = -6 \end{cases}$$

La méthode de Jacobi

Problème:

Soit le système d'équations linéares composé de n équation et n inconnus

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 ... + a_{2n}x_n = b_2 \\ . \\ . \\ a_{n1}x_1 + a_{n2}x_2 + ... + a_{nn}x_n = b_n \end{cases}$$

Les inconnes soit $x_1, x_2, ..., x_n$.

Les gradeurs $a_{11}, a_{12}, , ..., a_{nn}.$ et $b_1, b_2, ..., b_n.$ sont constantes connues.

- Trouver par la méthode de **Jacobi** la valeur de vecteur inconu X tel que $X = [x_1, x_2, ..., x_n]$, en admettant une erreur admissible ε.

Utiliser le le langage de programmation Matlab pour faire les calculs.

Principe de la méthode

3- Algothme de résolution

la méthode de *Jacobie, est une méthode itérative de résolution d'un système d'équatuions lineaires de la forme* $AX = b$ *pour cela, on utilise une suite* $X^{(k)}$ *qui converge vers le point fixe* X *qui sont la solution du système d'équations linéares.*

$$\begin{cases} x_1^{(k+1)} = \frac{1}{a_{11}} \left[b_1 - a_{12}x_1^{(k)} - ... - a_{1n}x_n^{(k)} \right] \\ x_2^{(k+1)} = \frac{1}{a_{22}} \left[b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)} ... - a_{2n}x_n^{(k)} \right] \\ . \\ . \\ x_n^{(k+1)} = \frac{1}{a_{nn}} \left[b_n - a_{1n}x_1^{(k)} - ... - a_{1n-1}x_{n-1}^{(k)} \right] \end{cases}$$

L'algorithme

1. The matrix $A\,(n \times n)$ must be non-singular and symétrical;

for $k = 1$ to n

$\det A\,(1:k) > 0$

enf for

2. calculate matrices Aoff and D:

$$D = diag\,(diag\,(A))$$

$$Aoff = A - D$$

3. calculatela the solution X:

for $k = 1$ to max

$x_{k+1} = inv\,(D)\,(b - Aoff \quad x_k)\,;$

if norm $(x_k - x_{k+1}) < tol$

fin itérations;

if not

$$x_k = x_{k+1}$$

end if

end for

$xsol = x_k$

Application

Ecrire un programme MATLAB qui implémente la méthode numérique de résolution numérique des systèmes d'équations linéaires (la méthode de Jacobi) et appliquer cette méthode pour trouver la solution des systèmes d'équations linéaires suivants:

$$\begin{cases} 5x_1 - 2x_2 + x_3 - x_4 = 4 \\ x_1 + 3x_2 - 2x_4 = 2 \\ 2x_1 + x_2 - 7x_3 + 2x_4 = 5 \\ 2x_1 + x_2 + 3x_3 - 6x_4 = -9 \end{cases}$$

Programmation sous Matlab

Programme principal

$$
\begin{aligned}
& clc \\
& clear \\
& fo\,\text{randmat long} \\
A &= [5\ \text{-}2\ 1\ \text{-}1;\ 1\ 3\ 0\ \text{-}2;\ 2\ 1\ \text{-}7\ 2;\ 2\ 1\ \text{-}7\ 2;\ 2\ 1\ 3\ \text{-}6]\,; \\
B &= [4\ ;\ 2;\ 5;\ \text{-}9]\,; \\
[4\ ;\ 2;\ 5;\ \text{-}9] &= MyJacobiFun(A, B, 1E-4)
\end{aligned}
$$

Sous programme

$$
\begin{aligned}
fonction\,[x, m] &= MyJacobiFun(A, B, EPS \\
[M, N] &= \text{Si}\,ze\,(A)\,; \\
if\ M..... &= N \\
&\quad disp\,('\text{pas de solution, Il faut introduire une matrice carrère}) \\
&\quad retun \\
&\quad end \\
D &= \det(A) \\
if\ D &= 0 \\
&\quad disp\,(!\ \text{pas de solution, déterminant égale à zéro!}') \\
&\quad retun \\
&\quad end \\
for\ i &= 1; N \\
sommeI &= sum\,(absA\,(1; i, i-1)) + sum\,(absA\,(i, i+1N))\,; \\
sommeJ &= sum\,(absA\,(1; i-1)) + sum\,(absA\,(i+1N, i))\,; \\
&\quad if abs\,(i, i)\,(some\ I)\,pr\ absA\,(i, 1)\,some \\
&\quad disp('la\ \text{matrice n'est pas à disgonale dominate!}) \\
&\quad end \\
x &= \\
&\quad 1,999913568491538 \\
&\quad 1,99871859424150 \\
&\quad 0,999863700180910 \\
&\quad 2,999826040588216 \\
m &= \\
&\quad 25
\end{aligned}
$$

3- Algothme de résolution

La méthode de Gass-Siedlel.

Problème:

Soit le système d'équations linéares composé de n équation et n inconnus

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2... + a_{2n}x_n = b_2 \\ . \\ . \\ a_{n1}x_1 + a_{n2}x_2 + ... + a_{nn}x_n = b_n \end{cases}$$

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

$\left| X^{(k+1)} - X^{(k)} \right| \leq \varepsilon$

Les inconnes soit $x_1, x_2, ..., x_n$.

Les gradeurs $a_{11}, a_{12}, , ..., a_{nn}.$ et $b_1, b_2, ..., b_n.$ sont constantes connues.

- Trouver par la méthode de **Jacobi** la valeur de vecteur inconu X tel que $X = [x_1, x_2, ..., x_n]$, en admettant une erreur admissible ε.

Utiliser le le langage de programmation Matlab pour faire les calculs.

La forme matricielle

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2... + a_{2n}x_n = b_2 \\ . \\ . \\ a_{n1}x_1 + a_{n2}x_2 + ... + a_{nn}x_n = b_n \end{cases} \iff \begin{pmatrix} a_{11} & a_{12} & . & a_{1n} \\ a_{21} & a_{22} & . & a_{2n} \\ . & . & . & . \\ . & . & . & . \\ a_{n1} & a_{n2} & . & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ . \\ . \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ . \\ . \\ b_n \end{pmatrix}$$

$$\iff [a_{ij}]\,[x_i] = [b_i] \iff AX = b$$

La notation i indique la ligne de la matrice la notation j indique la colonne de la

matrice A ou des vecteurs X et B.

La notation j indique la ligne de la matrice la notation j indique la colonne de la matrice A.

2) La matice augmenté

La matice augmentée est une forme plus simplifiée su système matriciel, on préfée l'utiliser pour représenter le système d'équation linéaires pendant les étapes de calculs avec cette matrice augmentée, on ne mentionne plus le vecteur X, telle que

3- Algothme de résolution

la méthode de Gass-Siedlel, *est une méthode itérative de résolution d'un système d'équatuions linéaires de la forme* $AX = b$ *pour cela, on utilise une suite* $X^{(k)}$ *qui converge vers le point fixe* X *qui sont la solution du système d'équations linéares.*

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 ... + a_{2n}x_n = b_2 \\ . \\ . \\ a_{n1}x_1 + a_{n2}x_2 + ... + a_{nn}x_n = b_n \end{cases}$$

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left[b_i - \sum_{i=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right]; \quad \left| X^{(m+1)} - X^{(m)} \right| \leq \varepsilon.$$

L'algorithme

1. La matrice A doit être non-singulière et symétrique;

2. La matrice $A\,(n \times n)$ doit être non-singulière et symétrique;

for $k = 1$ to n

$\det A\,(1 : k) > 0$

enf for

3. calculer les matrices Aoff et D:

$$D = diag\,(diag\,(A))$$

$$Aoff = A - D$$

4. calculer la solution X:

for $k = 1$ to max

$x_{k+1} = inv\,(D - E)\,(b - Aoff\ x_k)\,;$

If norm $(x_k - x_{k+1}) < tol$

fin itérations;

if not

$x_k = x_{k+1}$

end if

end for

$xsol = x_k$

Application

Ecrire un programme MATLAB qui implémente la méthode numérique de résolution numérique des systèmes d'équations linéaires (la méthode de Gass-Siedlel.) et appliquer cette méthode pour trouver la solution des systèmes d'équations linéaires suivants:

$$\begin{cases} 5x_1 - 2x_2 + x_3 - x_4 = 4 \\ x_1 + 3x_2 - 2x_4 = 2 \\ 2x_1 + x_2 - 7x_3 + 2x_4 = 5 \\ 2x_1 + x_2 + 3x_3 - 6x_4 = -9 \end{cases}$$

Programmation sous Matlab

Programme principal

$$
\begin{aligned}
& clc \\
& clear \\
& fo\,\text{randmat long} \\
A &= [5\ -21-1;130-2;21-72;21-72;213-6]; \\
B &= [4\ ;2;5;-9]; \\
[4\ ;2;5;-9] &= MyGass\,\text{Si}\,edlelFun(A,B,1E-4)
\end{aligned}
$$

Sous Programme

$$
\begin{aligned}
fonction\,[x,m] &= MyGass\,\mathrm{Si}\,edlelFun(A,B,EPS\\
[M,N] &= \mathrm{Si}\,ze\,(A)\,;\\
if\ M..... &= N\\
&\quad disp\,('\text{pas de solution, Il faut introduire une matrice carrère})\\
&\quad retun\\
&\quad end\\
D &= \det(A)\\
if\ D &= 0\\
&\quad disp\,(!\text{ pas de solution, déterminant égale à zéro!'})\\
&\quad retun\\
&\quad end\\
for\ i &= 1;N\\
sommeI &= sum\,(absA\,(1;i,i-1)) + sum\,(absA\,(i,i+1N))\,;;\\
&\quad ifabs\,(i,i) < some\ I\\
&\quad disp('\text{la matrice n'est pas à disgonale dominate!})\\
&\quad end\\
&\quad end\\
x &= zeros\,(N,1)\\
m &= 0\\
erreus &= Eps+1;\\
while\ erreur &> Eps\\
x &=\\
&\quad 1,999913568491538\\
&\quad 1,99871859424150\\
&\quad 0,999863700180910\\
&\quad 2,999826040588216\\
m &=\\
&\quad 14
\end{aligned}
$$

Bibliography

[1] SOUDANI AZEDDINE, Calcul numériques, Partie I: Cours d'analyse numerique OPU, (2015)

[2] CASTEYDE, Cours de C/C^{++}", Copyright, (2005).

[3] CORDE ET A. FOUILLOUX, Langage Fortran, Support de cours, IDRIS,(2010).

[4] D'ANFRAY,''Fortran 77", Université Paris XIII, (1998).

[5] DJEBLI & H. DJELOUAH, Initiation à MATLAB, OPU, (2013).

[6] DUKKIPATI, MATLAB, an introduction with applications, New Age International Publishers, India, (2010).

[7] DELANNOY, C^{++} pour les programmeurs C", 6ème Ed., Eyrolles, Paris, (2004).

[8] HAHN, Introduction to Fortran 90 for scientists and engineers", Capetown University, South Africa, (1993).

[9] JEDRZEIJEWSKI, Introduction aux méthodes numériques, 2ème Ed., Springer, France, (2005).

[10] HOFFMAN, Numerical methods for engineers and scientists, 2nd Ed, Marcel Dekker, USA, (2001).

[11] A. QUARTERONI, Méthodes numériques, algorithmes, analyse et appl., Springer, Italie, (2004).

[12] Mustapha lakrib, Cours d'analyse numerique OPU, (2003).

[13] Karima MEBARKI, Analyse Numérique, Cours, $2^{éme}$ année licence mathématiques., Université de Béjaia.

[14] WOODFORD and C. Phillips, Numerical methods with worked examples: MATLAB edition, 2nd Ed. Springer Ltd, (2013).