

*Université Abdel hamid Ibn Badis Mostaganem*

*Faculté des sciences exactes et de l'informatique*

*Département de physique*

*Année universitaire 2021/2022*

# *Polycopié de Cours et TP de Méthodes Numériques*

*Préparé par : Belhouari Aissa et Belayachi Cherifa*

*Expertise :*

*Expert 1 : Dr. Rahal Wassila Leila– Université de Mostaganem*

*Expert 2 : Prof. Belhamiti Omar département de mathématiques - Université de Mostaganem*

---

# Table des Matières

<b>INTRODUCTION .....</b>	<b>8</b>
<b>Chapitre 1 RESOLUTION DES EQUATIONS NON LINEAIRES .....</b>	<b>10</b>
<b>1.1 Introduction .....</b>	<b>10</b>
<b>1.2 Localisation des Racines par la « Méthode de Tabulation » .....</b>	<b>10</b>
1.2.1 Définition .....	10
1.2.2 L'algorithme .....	10
1.2.3 Programme de la « méthode de Tabulation » .....	12
<b>1.3 Méthode de Bissection (Dichotomie).....</b>	<b>14</b>
1.3.1 Définition .....	14
1.3.2 L'algorithme .....	14
1.3.3 Programmes de la « Méthode de Bissection – Dichotomie ».....	16
<b>1.4 Méthode du point Fixe .....</b>	<b>18</b>
1.4.1 Définition .....	18
1.4.2 Théorème.....	18
1.4.3 Algorithme .....	18
1.4.4 Programme de la « méthode du Point Fixe » .....	20
<b>1.5 Méthode de Newton – Raphson.....</b>	<b>21</b>
1.5.1 Définition .....	21
1.5.2 L'algorithme.....	22
1.5.3 Programme de la « Méthode de Newton – Raphson » .....	23
<b>1.6 Racines Multiples .....</b>	<b>23</b>
1.6.1 Définition .....	23
1.6.2 Théorème.....	23
<b>1.7 Cas de fonction Polynômiale .....</b>	<b>23</b>
1.7.1 Evaluation de Polynôme (Schéma de HORNER) .....	23
1.7.1.1 Programme du « Schéma de Horner ».....	24
1.7.2 Division de Polynôme .....	24
1.7.3 Séparation et Calcul du Nombre des Racines .....	24
1.7.3.1 Règle de DESCARTES.....	24
1.7.3.2 Théorème de STURM .....	25
1.7.3.2.1 Programme du « Théorème de STIRM » .....	25
<b>Chapitre 2 SYSTEME D'EQUATIONS LINEAIRES .....</b>	<b>27</b>
<b>2.1 Introduction .....</b>	<b>27</b>
<b>2.2 Méthodes d'Elimination de Gauss (Triangularisation) .....</b>	<b>27</b>

---

2.2.1	Définition .....	27
2.2.2	L'algorithme.....	28
2.2.3	Généralisation de la Triangularisation pour un système $N \times N$ .....	31
2.2.4	Programme de la méthode « Elimination de Gauss ».....	35
2.2.4.1	Système à $3 \times 3$ .....	35
2.2.4.2	Système à $N \times N$ .....	36
<b>2.3</b>	<b>Décomposition LU (factorisation).....</b>	<b>36</b>
2.3.1	Définition .....	36
2.3.2	Théorème.....	36
2.3.3	L'algorithme.....	37
2.3.4	Programme de la méthode de « Décomposition LU » .....	40
<b>2.4</b>	<b>Méthode Itérative de Jacobi.....</b>	<b>41</b>
2.4.1	Définition .....	41
2.4.2	L'algorithme.....	41
2.4.3	Théorème.....	42
2.4.4	Programme de la « Méthode Itérative de Jacobi » .....	43
<b>Chapitre 3</b>	<b>INTEGRATIONS NUMERIQUES .....</b>	<b>46</b>
<b>3.1</b>	<b>Introduction.....</b>	<b>46</b>
<b>3.2</b>	<b>Définition « Intégrale » .....</b>	<b>46</b>
<b>3.3</b>	<b>Méthode des Rectangles.....</b>	<b>46</b>
3.3.1	Définition .....	46
3.3.2	Théorème.....	47
3.3.3	L'algorithme.....	48
3.3.4	Programme de la « Méthode des Rectangles ».....	50
<b>3.4</b>	<b>Méthode des Trapèzes.....</b>	<b>50</b>
3.4.1	Définition .....	50
3.4.2	Théorème.....	51
3.4.3	L'algorithme.....	51
<b>3.5</b>	<b>Méthode de Simpson .....</b>	<b>53</b>
3.5.1	Définition .....	53
3.5.2	Théorème.....	54
3.5.3	L'algorithme.....	54
<b>Chapitre 4</b>	<b>PROBLEME AUX VALEURS PROPRES .....</b>	<b>58</b>
<b>4.1</b>	<b>Introduction.....</b>	<b>58</b>
<b>4.2</b>	<b>Méthode Polynomiale.....</b>	<b>59</b>

---

<b>4.3</b>	<b>La Méthode de FADEEV – LEVERRIER</b> .....	<b>60</b>
4.3.1	Définition .....	60
4.3.2	L’algorithmme.....	61
4.3.3	Programme de la « Méthode de Fadeev – Leverrier » .....	62
<b>4.4</b>	<b>Méthode de Jacobi</b> .....	<b>64</b>
4.4.1	Définition .....	64
4.4.2	Programme de la « Méthode de Jacobi » .....	66
<b>Chapitre 5</b>	<b>INTERPOLATION POLYNOMIALE</b> .....	<b>70</b>
<b>5.1</b>	<b>Interpolation – Définition</b> .....	<b>70</b>
<b>5.2</b>	<b>Interpolation polynomiale</b> .....	<b>70</b>
<b>5.3</b>	<b>Corollaire</b> .....	<b>70</b>
<b>5.4</b>	<b>Polynôme de LAGRANGE</b> .....	<b>70</b>
5.4.1	L’expression.....	71
5.4.2	Théorème.....	71
<b>5.5</b>	<b>Polynôme d’Interpolation de Newton (Interpolation de Newton)</b> .....	<b>71</b>
<b>5.6</b>	<b>Méthode des Coefficients Indéterminés</b> .....	<b>72</b>
<b>Chapitre 6</b>	<b>METHODES DES ELEMENTS FINIS</b> .....	<b>76</b>
<b>6.1</b>	<b>Introduction</b> .....	<b>76</b>
<b>6.2</b>	<b>Exprimer le problème sous forme variationnelle ou intégrale</b> .....	<b>76</b>
<b>6.3</b>	<b>Maillage ou discrétisation</b> .....	<b>77</b>
<b>6.4</b>	<b>Choix des fonctions de formes</b> .....	<b>77</b>
<b>Chapitre 7</b>	<b>DERIVATION NUMERIQUE</b> .....	<b>85</b>
<b>7.1</b>	<b>Introduction</b> .....	<b>85</b>
<b>7.2</b>	<b>Définition de la méthode des différence finies</b> .....	<b>85</b>
7.2.1	Introduction .....	85
<b>7.3</b>	<b>Opérateurs de différences finies</b> .....	<b>85</b>
7.3.1	Opérateur différence avant .....	85
7.3.2	Opérateur différence arrière .....	85
7.3.3	Opérateur différence centrée .....	85
<b>7.4</b>	<b>Relation avec la dérivée d’une fonction</b> .....	<b>86</b>
7.4.1	Théorème 1.....	86
7.4.2	Théorème 2.....	86
7.4.3	Définition .....	86
<b>7.5</b>	<b>Programme de la méthode de « Dérivation Numérique »</b> .....	<b>87</b>
<b>Chapitre 8</b>	<b>RESOLUTION NUMERIQUE DES EQUATIONS DIFFERENTIELLES</b>	<b>90</b>

---

<b>8.1</b>	<b>Définition.....</b>	<b>90</b>
8.1.1	Equation différentielle ordinaire .....	90
8.1.2	Existence et Unicité de la solution .....	90
<b>8.2</b>	<b>Méthode de Série de Taylor.....</b>	<b>90</b>
8.2.1	Formulation .....	90
<b>8.3</b>	<b>Méthode de Piccard.....</b>	<b>91</b>
8.3.1	Formulation .....	91
<b>8.4</b>	<b>Application de la « Méthode des Différences Finies ».....</b>	<b>91</b>
8.4.1	L’algorithme.....	91
8.4.2	Méthode d’Euler.....	92
8.4.2.1	Définition .....	92
<b>8.5</b>	<b>Equation Différentielle du 1<sup>er</sup> Ordre .....</b>	<b>92</b>
8.5.1	Programme correspondant à une Equation Différentielle du 1er Ordre – « Méthode d’Euler »	93
<b>8.6</b>	<b>Equation Différentielle du 2<sup>nd</sup> Ordre.....</b>	<b>93</b>
8.6.1	Problème avec Conditions Initiales .....	93
8.6.1.1	Recherche du schéma récursif.....	94
8.6.2	Programme correspondant à une Equation du 2 <sup>nd</sup> Ordre avec Conditions Initiales .....	96
8.6.3	Problème aux limites .....	96
8.6.3.1	Théorème.....	97
8.6.3.2	Méthode du Tir (problème avec conditions aux limites).....	97
<b>8.7</b>	<b>Equations aux Dérivées Partielles.....</b>	<b>98</b>
8.7.1	Résolution d’Equations aux Dérivées Partielles.....	99
<b>Chapitre 9 METHODES DE MONTE – CARLO.....</b>		<b>110</b>
<b>9.1</b>	<b>Définition.....</b>	<b>110</b>
<b>9.2</b>	<b>Rappel sur les notions de base de probabilité des variables aléatoires .....</b>	<b>110</b>
9.2.1	Définition .....	110
9.2.2	Variables aléatoires discrètes .....	110
9.2.2.1	L’espérance .....	111
9.2.2.1.1	Définition .....	111
9.2.2.2	La variance .....	112
9.2.3	Variables aléatoires continues .....	112
9.2.3.1	L’espérance .....	112
9.2.3.2	La distribution uniforme.....	113
9.2.3.3	La loi de Poisson .....	113
9.2.3.4	La loi normale ou la loi de Laplace-Gauss.....	113

---

9.2.3.5	Cas de plusieurs variables aléatoires .....	113
<b>9.3</b>	<b>Principe de la méthode Monte – Carlo.....</b>	<b>113</b>
9.3.1	La loi des grands nombres.....	113
9.3.2	Théorème de la limite centrale .....	114
<b>9.4</b>	<b>Intégration par la Méthode Monte – Carlo.....</b>	<b>114</b>
9.4.1	Génération de variables aléatoires.....	115
<b>9.5</b>	<b>Algorithme « Génération de Nombres Aléatoires » dans l’Intervalle 0, 1 –</b>	
	<b>« Algorithme de Congruence ».....</b>	<b>116</b>
9.5.1	Fonction et Subroutine de génération de nombre aléatoire en Fortran.....	116
9.5.1.1	Syntaxe.....	116
9.5.2	Programme de la méthode « Génération des nombres aléatoires » dans [0 ; 1].....	117
9.5.3	Programme de « l’Algorithme de Congruence ».....	117
<b>9.6</b>	<b>Algorithme « Calcul d’une Intégrale ».....</b>	<b>118</b>
9.6.1	Programme de la méthode « Monte – Carlo » pour le calcul d’intégrale.....	118

---

# **INTRODUCTION**

---

# INTRODUCTION

La question qui se pose est : « **Pourquoi Les Méthodes Numériques En Physique ?** ». Les lois qui régissent les phénomènes physiques sont exprimées par des équations mathématiques (équations différentielles, équations intégrales, etc.). Étudier le phénomène consiste souvent à trouver les solutions analytiques à ces équations chaque fois que c'est possible. Malheureusement trouver une solution analytique exacte n'est pas toujours possible. Le physicien introduit des approximations indispensables à la résolution du problème sans autant réduire la généralité du phénomène. Si non il fait recours aux méthodes numériques qui deviennent l'unique issue. Les méthodes numériques sont toutes les techniques de calcul qui permettent de résoudre de manière exacte ou souvent approchée un problème donné. L'analyse numérique est une discipline des mathématiques qui s'intéresse à la mise en pratique de ces méthodes permettant ainsi de résoudre par le calcul purement numérique certains problèmes difficiles où impossible à résoudre.

Analytiquement. Le calcul numérique est un ensemble de calculs qui sont réalisés sur un système informatique. Contrairement à la résolution exacte et analytique d'un problème où la solution est exempte d'erreurs, la résolution numérique conduit souvent à des solutions entachées d'erreurs provenant de 3 sources principales :

## **a. Erreurs de Modélisation**

Modéliser un phénomène revient à produire un ensemble d'équation mathématiques qui le décrivent. Pour réussir à les résoudre il est souvent nécessaire d'introduire des approximations qui sont à l'origine des erreurs de modélisation.

## **b. Erreurs de Représentation sur Ordinateur**

Le système binaire (base 2) est à la base de la représentation des nombres sur ordinateur. L'unité fondamentale de l'information est le bit qui peut prendre les valeurs 0 ou 1. Les données sont représentées dans la mémoire de l'ordinateur par des mots (case) un ensemble de  $n$  bits (8,16,32..) l'intervalle des nombres représentables est limité ce qui induit des erreurs d'arrondi.

## **c. Erreurs de Troncature**

Ces erreurs sont dues à la représentation finie de processus infinie par exemple le développement de série de Taylor des fonctions élémentaires utilisé pour leur calcul. On est obligé de s'arrêter à un certain ordre du développement.



# **Chapitre 1**

## **RESOLUTION DES EQUATIONS NON LINEAIRES**

---

## Chapitre 1 RESOLUTION DES EQUATIONS NON LINEAIRES

### 1.1 Introduction

En physique comme en d'autres branches de la science, on est souvent confronté à des équations de la forme  $f(x) = 0$ . Une valeur  $r$  solution de l'équation  $f(x) = 0$  c.à.d.  $f(r) = 0$  est appelée racine ou zéro de l'équation. La recherche de la forme analytique exacte de la solution est souvent difficile. Déjà les polynômes  $p(x)$  (les fonctions les plus simples) de degré  $n > 5$  Galois a démontré qu'il n'existe pas de méthodes exactes analytiques pour le calcul des solutions. Il faut donc recourir aux méthodes numériques.

### 1.2 Localisation des Racines par la « Méthode de Tabulation »

Généralement les racines sont localisées soit par une méthode graphique, les racines sont alors les abscisses des points d'intersection du graphe avec l'axe des  $x$ . Dans certains cas où on peut transformer le problème  $f(x) = 0$  en un problème équivalent  $g(x) = h(x)$ , les racines sont les abscisses des points d'intersection des graphes de  $g$  et  $h$ .

#### 1.2.1 Définition

Cette méthode de tabulation est basée sur le théorème des valeurs intermédiaires : si une fonction  $f$  est continue sur un intervalle  $[a, b]$  tel que  $f(a) \times f(b) < 0$  (change de signe) alors il existe au moins un  $c$  tel que  $f(c) = 0$ . La valeur de la fonction est calculée pour des points discrets  $x_i$   $i = 1, 2, \dots, n$  puis on détecte tous les intervalles  $[x_i, x_{i+1}]$  dans lesquelles la fonction change de signe comme ça on aura localisé ses racines.

#### 1.2.2 L'algorithme

A partir d'un certain intervalle  $[a, b]$  qu'on discrétise en points  $\{x_0; x_1; \dots; x_i; \dots; x_n\}$  tel que  $x_i = x_0 + i \times h$  avec  $x_0 = a, x_n = b$  et  $h$  est le pas. Pour chaque sous-intervalle, on effectue les produits suivants  $f(x_i) \times f(x_{i+1})$  c.à.d.  $\{(x_0; f(x_0)), (x_1; f(x_1)), \dots, (x_i; f(x_i)), \dots, (x_n; f(x_n))\}$  et à chaque fois on teste son signe :

- a- Déterminer l'intervalle  $[a, b]$
- b- Choisir le pas  $h$
- c- Effectuer les calculs  $x_i; f(x_i); x_{i+1}; f(x_{i+1})$
- d- Tester le produit  $f(x_i) \times f(x_{i+1})$  si c'est négatif  $\Rightarrow \exists$  au moins une racine  $r \in [x_i; x_{i+1}]$
- e- Si non, continuer avec les autres valeurs de l'intervalle  $[x_{i+2}; \dots; x_n]$

#### Exemple

Essayer de localiser les racines pour les équations suivantes :

$$f(x) = 8x^3 - 20x^2 - 2x + 5 = 0 \quad \text{et} \quad g(x) = 3^x - 3x - 2 = 0$$

Pour localiser les racines d'une fonction définie sur  $\mathbb{R}$ , il est préférable de tracer le graphe de cette fonction à l'aide d'un logiciel et réduire au maximum l'intervalle dans lequel il faut chercher la racine. Le graphe permettra d'avoir une vision plus précise sur la position de la racine dans  $\mathbb{R}$ .

#### Solution

- a. **Fonction  $f(x)$**  : On commence par tracer le graphe de  $f(x)$

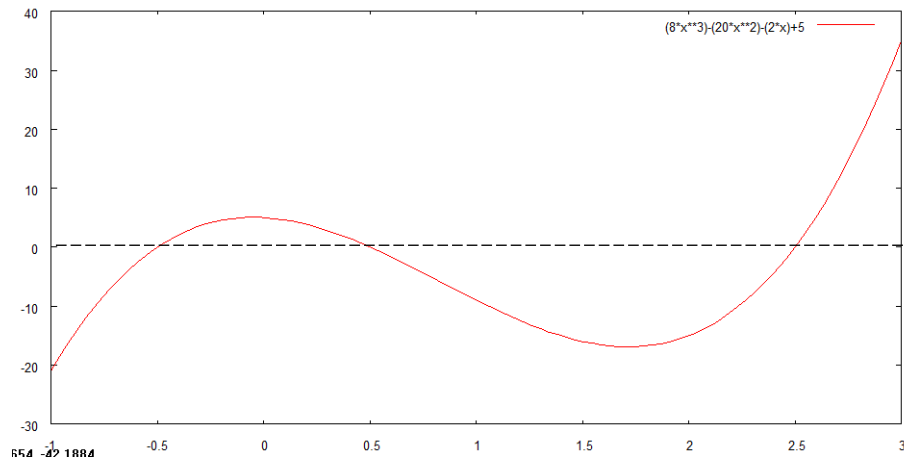


Figure 1.1 : Graphe de la fonction  $f(x)$

On remarque d’après le graphe que la fonction possède trois (03) racines situées dans l’intervalle  $[-1 ; 3]$ . On réduit l’intervalle de chacune de ces trois racines pour procéder à une résolution de l’équation par une des méthodes numériques dédiées à cela. La localisation (réduction de l’intervalle) se fait par la méthode de Tabulation.

**Calcul manuel** : Divisons l’intervalle  $[-1 ; 3]$  en 20 points comme exemple concret pour voir le calcul manuel.

- On détermine le pas de calcul  $h : h = \frac{3-(-1)}{20} = 0,2$
- Les points de discrétisation  $(x_i ; f(x_i)) : \begin{cases} x_i = x_0 + i \times h \\ f(x_i) = 8x_i^3 - 20x_i^2 - 2x_i + 5 \end{cases}$
- Test du produit  $f(x_i) \times f(x_{i+1})$

Le tableau 1.1 résume les résultats de calcul.

Tableau 1.1 : Résultats de calcul appliquant la tabulation à la fonction  $f(x)$

$x_i$	$f(x_i)$	$x_i$	$f(x_i)$
-1	-21	1,2	-12,376
-0,8	-10,296	1,4	-15,048
-0,6	<b>-2,728</b>	1,6	-16,632
-0,4	<b>2,088</b>	1,8	-16,744
-0,2	4,536	2	-15
0	5	2,2	-11,016
0,2	3,864	<b>2,4</b>	<b>-4,408</b>
<b>0,4</b>	<b>1,512</b>	<b>2,6</b>	<b>5,208</b>
<b>0,6</b>	<b>-1,672</b>	2,8	18,216
<b>0,8</b>	-5,304	3	35
1	-9		

Le calcul manuel nous indique que les racines se situent respectivement dans :

$$r_1 \in [-0,6 ; -0,4] \quad ; \quad r_2 \in [0,4 ; 0,6] \quad ; \quad r_3 \in [2,4 ; 2,6]$$

- b. Fonction  $g(x)$  : Les mêmes étapes s’appliquent pour cet exemple

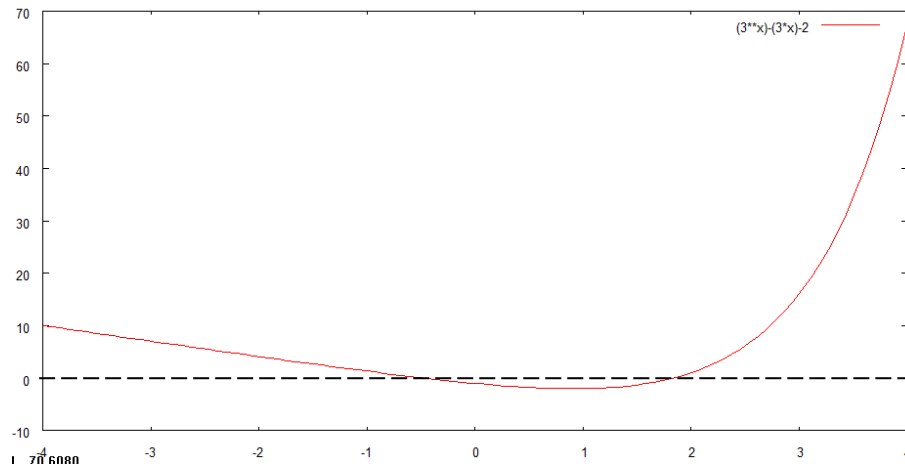


Figure 1.2 : Graphe de la fonction  $g(x)$

On remarque d’après le graphe (figure 1.2) que la fonction  $g(x)$  possède deux (02) racines situées dans l’intervalle  $[-4 ; 4]$ . On procède de même pour la localisation des racines.

**Calcul manuel :** Divisons l’intervalle  $[-4 ; 4]$  en près de 40 points comme exemple concret pour voir le calcul manuel.

- On détermine le pas de calcul  $h : h = \frac{4 - (-4)}{40} = 0,2$
- Les points de discrétisation  $(x_i ; g(x_i)) : \begin{cases} x_i = x_0 + i \times h \\ g(x_i) = 3^{x_i} - 3x_i - 2 \end{cases}$
- Test du produit  $g(x_i) \times g(x_{i+1})$

Le tableau 1.2 résume les résultats de calculs.

Tableau 1.2 : Résultats de calcul appliquant la tabulation à la fonction  $g(x)$

$x_i$	$g(x_i)$	$x_i$	$g(x_i)$	$x_i$	$g(x_i)$	$x_i$	$g(x_i)$
-4	10,01	-1,8	3,54	0,2	-1,35	2,4	4,77
-3,6	8,82	-1,6	2,97	0,6	-1,87	2,6	7,60
-3,4	8,22	-1,4	2,41	0,8	-1,99	2,8	11,27
-3,2	7,63	-1,2	1,87	1	-2	3	16
-3	7,04	-1	1,33	1,2	-1,86	3,2	22,03
-2,8	6,45	-0,8	0,82	1,4	-1,54	3,4	29,70
-2,6	5,86	<b>-0,6</b>	<b>0,32</b>	1,6	-1,00	3,6	39,40
-2,4	5,27	<b>-0,4</b>	<b>-0,16</b>	<b>1,8</b>	<b>-0,18</b>	3,8	51,62
-2,2	4,69	-0,2	-0,60	<b>2</b>	<b>1</b>	4	67
-2	4,11	0	-1	2,2	2,61		

Le calcul manuel nous indique que les deux racines  $r_1$  et  $r_2$  sont situées dans les intervalles :  $[-0,6 ; -0,4]$  et  $[1,8 ; 2]$  respectivement.

### 1.2.3 Programme de la « méthode de Tabulation »

#### a) Fonction $f(x)$

```

Program Tabulation
implicit none
real, dimension(10000000):: x
real:: f, h, a, b
integer:: i

!---Entrée des données
write(*,*) "Entrez l'intervalle [a,b] et le pas h"
Read(*,*) a, b, h

x(0)=a
i=0
!---Discrétisation de l'intervalle [a;b]
Do
i=i+1
x(i)=x(0)+i*h

!---Test du produit f(x(i))*f(x(i+1))
if((f(x(i-1))*f(x(i)))<0) exit
Enddo

!---Affichage des résultats
write(*,*) "la racine se situe dans l'intervalle"
write(*,*) "[", x(i-1), ";", x(i), "]"

End

!---La Fonction
Function f(x)
f=(8*x**3)-(20*x**2)-(2*x)+5
Return
Endfunction

```

```

F:\Software Setups\Fortran9...
Entrez l'intervalle [a,b] et le pas h
-1
3
0.2
la racine se situe dans l'intervalle
[ -6.000000E-01; -4.000000E-01]
Press any key to continue

```

Figure 1.3 : Programme FORTRAN de la Méthode de Tabulation pour la fonction  $f(x)$

### Remarque

L'intervalle  $[-1 ; 3]$  de  $f(x)$  contient les trois racines de cette fonction. Or le programme n'a détecté que l'intervalle contenant la première racine (figure 1.3). D'où l'utilité de chercher chaque racine individuellement, c.à.d. compiler le programme pour les trois différents intervalles des racines comme il est illustré sur la figure 1.4.

```

Program Tabulation
implicit none
real, dimension(10000000):: x
real:: f, h, a, b
integer:: i

!---Entrée des données
write(*,*) "Entrez l'intervalle [a,b] et le pas h"
Read(*,*) a, b, h

x(0)=a
i=0
!---Discrétisation de l'intervalle [a;b]
Do
i=i+1
x(i)=x(0)+i*h

!---Test du produit f(x(i))*f(x(i+1))
if((f(x(i-1))*f(x(i)))<0) exit
Enddo

!---Affichage des résultats
write(*,*) "la racine se situe dans l'intervalle"
write(*,*) "[", x(i-1), ";", x(i), "]"

End

!---La Fonction
Function f(x)
f=(8*x**3)-(20*x**2)-(2*x)+5
Return
Endfunction

```

```

F:\Software Setups\Fortran9...
Entrez l'intervalle [a,b] et le pas h
-1
0
0.2
la racine se situe dans l'intervalle
[ -6.000000E-01; -4.000000E-01]

F:\Software Setups\Fortran9...
Entrez l'intervalle [a,b] et le pas h
0
1
0.2
la racine se situe dans l'intervalle
[ 4.000000E-01; 6.000000E-01]

F:\Software Setups\Fortran9...
Entrez l'intervalle [a,b] et le pas h
2
3
0.2
la racine se situe dans l'intervalle
[ 2.400000; 2.600000]

```

Figure 1.4 : Compilation du programme de Tabulation pour les trois racines

### b) Fonction $g(x)$

```

Program Tabulation
implicit none
real, dimension(10000000):: x
real:: g, h, a, b
integer:: i

!---Entrée des données
write(*,*) "Entrez l'intervalle [a,b] et le pas h"
Read(*,*) a, b, h

x(0)=a
i=0
!---Discretisation de l'intervalle [a;b]
Do
i=i+1
x(i)=x(0)+i*h

!---Test du produit g(x(i))*g(x(i+1))
if((g(x(i-1))*g(x(i)))<0) exit
Enddo

!---Affichage des résultats
write(*,*) "la racine se situe dans l'intervalle"
write(*,*) "[", x(i-1), ";", x(i), "]"

End

!---La Fonction
Function g(x)
g=(3**x)-(3*x)-2
Return
Endfunction

```

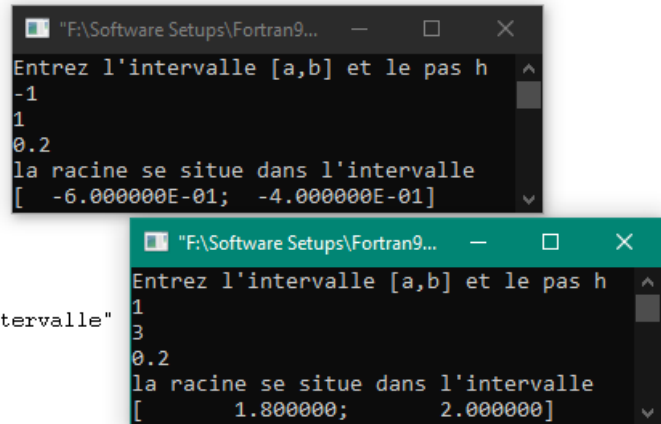


Figure 1.5 : Programme FORTRAN de la Méthode de Tabulation pour la fonction  $g(x)$

## 1.3 Méthode de Bisection (Dichotomie)

### 1.3.1 Définition

Cette méthode repose sur l'idée toute simple qu'une fonction continue sur un intervalle  $[a, b]$  qui change de signe doit obligatoirement couper l'axe des  $x$ . ceci est illustré mathématiquement par le théorème des valeurs intermédiaires énoncé ci-dessus. Si de plus  $f$  est monotone alors  $r$  est unique. On a une fonction  $f$  continue sur  $[a, b]$  tel que  $f(a) \times f(b) < 0$ . Pour déterminer la racine  $r$  supposée unique avec une certaine précision  $\epsilon$  on calcule le milieu  $c = (b + a)/2$  si  $f(a) \times f(c) < 0$  alors  $r \in [a, c]$  si non  $r \in [c, b]$  on répète la procédure avec le nouvel intervalle et ainsi de suite, on refait la même procédure en continuant le processus jusqu'à la  $n$ ème itération qui satisfait le critère d'arrêt  $L_n < \epsilon$  où  $L_n = (b_n - a_n)/2 = L/2^n$ . La racine  $r$  appartient à  $[a_n, b_n]$ . La valeur approchée de  $r$  est le centre de l'intervalle  $c_n = (a_n + b_n)/2$ , donc  $r = c_n \pm L_n$

On peut estimer le nombre d'itération nécessaire pour obtenir une certaine erreur absolue  $\epsilon$  par la relation :

$$\epsilon > L/2^n = (b - a)/2^n \quad (\text{I.1})$$

### 1.3.2 L'algorithme

Les étapes de l'algorithme peuvent être décrite comme suit :

- i.  $c = (a + b)/2$
- ii. Si  $f(a) \times f(c) < 0$  alors  $b = c$
- iii. Si  $f(c) \times f(b) < 0$  alors  $a = c$
- iv. Précision atteinte (critère d'arrêt) arrêt
- v. Si non retour à l'étape i.

La méthode est lente mais la convergence est sûre.

### Exemple 1

En utilisant la méthode de **Bissection**, calculez la racine de la fonction  $f(x) = x^3 - 4x + 1$  entre  $[1; 2]$  avec une erreur absolue de 0.02 et un nombre maximum d'itérations  $N_{max} = 10$ .

**Solution**

- i. Division de l'intervalle :  $c_1 = \frac{b+a}{2} = \frac{2+1}{2} = 1,5$
- ii. Test du produit  $f(a) \times f(c) < 0$  :  $\begin{cases} f(a) \times f(c) = (-2) \times (-1,63) > 0 \\ f(c) \times f(b) = (-1,63) \times (1) < 0 \end{cases}$
- iii. Localisation de la racine : On remarque que la racine se situe dans le demi-intervalle  $[c_1 ; b] \Rightarrow a = c_1$
- iv. Vérification de la précision (si  $\Delta I \leq \epsilon$ ) :  $\Delta I = \frac{b-a}{2} = \frac{2-1}{2} = 0,5 > \epsilon$
- v. Précision pas encore atteinte, on reprend les mêmes étapes avec le nouvel intervalle  $[c_1 ; b]$ .

On résume les résultats de calcul dans le tableau 1.3.

**Tableau 1.3 :** Résultats de calculs de la méthode de Bissection (exemple 1)

a	b	c	f(a)	f(b)	f(c)	f(a) × f(c)	f(c) × f(b)	$\Delta I = \frac{b-a}{2}$	$\epsilon$
1	2	1,50	-2	1	-1,63	3,25	<b>-1,63</b>	0,50	<b>0,02</b>
<b>1,50</b>	2	1,75	-1,63	1	-0,64	1,04	<b>-0,64</b>	0,25	
<b>1,75</b>	2	1,88	-0,64	1	0,09	<b>-0,06</b>	0,09	0,13	
<b>1,75</b>	1,88	1,81	-0,64	0,09	-0,30	0,19	<b>-0,03</b>	0,06	
<b>1,81</b>	1,88	1,84	-0,30	0,09	-0,11	0,03	<b>-0,01</b>	0,03	
<b>1,84</b>	1,88	<b>1,86</b>	-0,11	0,09	-0,01	0,00098	<b>-0,00084</b>	<b>0,02</b>	

On déduit de ce calcul que la racine de cette fonction est  $r \approx c \approx 1,86$

**Exemple 2**

En utilisant la méthode de **Bissection**, calculez la racine de la fonction  $f(x) = e^x - 2$  entre  $[0; 1,4]$  avec une erreur absolue de 0,01 et un nombre maximum d'itérations  $N_{max} = 10$

**Solution**

On suit les mêmes étapes suivis dans l'exemple précédent. Les résultats sont résumés dans le tableau 1.4. La précision est atteinte après 8 itérations où on obtient une précision de 0.005

On déduit de ce calcul que la racine de cette fonction est  $r \approx c \approx 0,69$

**Tableau 1.4 :** Résultats de calculs de la méthode de Bissection (exemple 2)

a	b	c	f(a)	f(b)	f(c)	f(a) × f(c)	f(c) × f(b)	$\Delta I = \frac{b-a}{2}$	$\epsilon$
<b>0,00</b>	1,40	0,70	-1,00	2,06	0,01	<b>-0,01</b>	0,03	0,70	<b>0,01</b>
<b>0,00</b>	0,70	0,35	-1,00	0,01	-0,58	0,58	<b>-0,01</b>	0,35	
<b>0,35</b>	0,70	0,53	-0,58	0,01	-0,31	0,18	<b>-0,004</b>	0,18	
<b>0,53</b>	0,70	0,61	-0,31	0,01	-0,15	0,05	<b>-0,002</b>	0,09	
<b>0,61</b>	0,70	0,66	-0,15	0,01	-0,07	0,01	<b>-0,001</b>	0,04	
<b>0,66</b>	0,70	0,68	-0,07	0,01	-0,03	0,00216	<b>-0,00041</b>	0,02	
<b>0,68</b>	0,70	0,69	-0,03	0,01	-0,01	0,00024	<b>-0,00011</b>	0,011	
<b>0,69</b>	0,70	<b>0,69</b>	-0,01	0,01	0,00	<b>-0,00002</b>	0,00004	<b>0,005</b>	

**Exemple 3**

On utilisera méthode de **Bissection** pour calculer les racines des fonctions  $f(x) = 8x^3 - 20x^2 - 2x + 5 = 0$  et  $g(x) = 3^x - 3x - 2 = 0$  de la partie « Méthode de Tabulation ». La figure 8 illustre le programme utilisé pour la résolution.

Les trois racines de  $f(x)$  sont :  $\begin{cases} r_1 \approx -0,40001 \\ r_2 \approx +0,59999 \\ r_3 \approx +2,59999 \end{cases}$  et les deux racines de  $g(x)$  sont :  $\begin{cases} r_1 \approx -0,46714 \\ r_2 \approx +1,83445 \end{cases}$

### 1.3.3 Programmes de la « Méthode de Bissection – Dichotomie »

#### i. Exemple 1

```
implicit none
doubleprecision :: a,b,c,largeur,epsilon
real :: f
write(*,*) "entrez les données a, b et epsilon"
read(*,*) a, b ,epsilon
5 c=(a+b)/2
largeur=(b-a)/2
!----Test de Précision
if(largeur> epsilon)then
!----Test de Produit f(a)*f(c)
if(f(a)*f(c)<0)then
b=c
goto 5
else
a=c
goto 5
endif
else
!----Affichage du Résultat
write(*,*) "la racine est", c
endif
end
!----Fonction
function f(x)
doubleprecision::x
f=(x**3)-(4*x)+1
return
endfunction
```

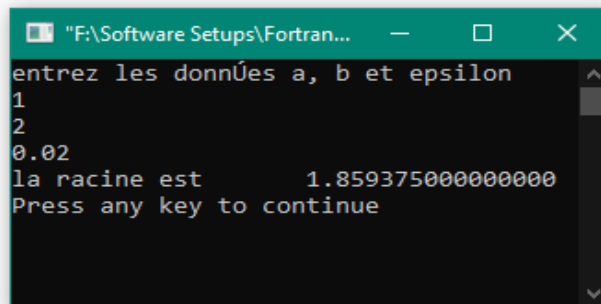


Figure 1.6 : Programme de l'exemple 1 avec résultat

#### ii. Exemple 2

```
implicit none
doubleprecision :: a,b,c,largeur,epsilon
real :: f
write(*,*) "entrez les données a, b et epsilon"
read(*,*) a, b ,epsilon
5 c=(a+b)/2
largeur=(b-a)/2
!----Test de Précision
if(largeur> epsilon)then
!----Test de Produit f(a)*f(c)
if(f(a)*f(c)<0)then
b=c
goto 5
else
a=c
goto 5
endif
else
!----Affichage du Résultat
write(*,*) "la racine est", c
endif
end
!----Fonction
function f(x)
doubleprecision::x
f=exp(x)-2
return
endfunction
```

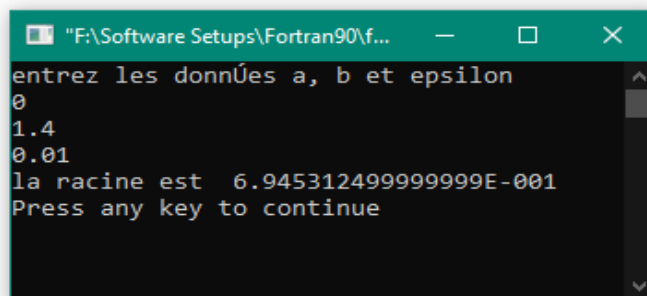


Figure 1.7 : Programme de l'exemple 2 avec résultat



Remarque

Le programme de cet exemple 2 est identique à celui du premier exemple. Il n'y a que la fonction  $f(x)$  dans la subroutine qui change (encerclée sur l'image de la figure 1.7).

iii. Exemple 3

```

Program bissection
implicit none
real:: a, b, c, r, e, f, m
!integer:: i

!----Entrée des données
Write(*,*) "Entrez l'intervalle [a, b] et e (epsilon)"
Read(*,*) a, b, e

!----Boucle
Do
10 c=(b+a)/2
!----Test
if (f(a)*f(c)<0) then
    b=c
    m=(b-a)/2
    if (m<=e) exit
    goto 10
else
    a=c
    m=(b-a)/2
    if (m<=e) exit
    goto 10
endif
enddo
r=(b+a)/2
Write(*,*) "la racine est r=", r

end

Function f(x)
f=(8*x**3)-(20*x**2)-(2*x)+5
Return
Endfunction
    
```

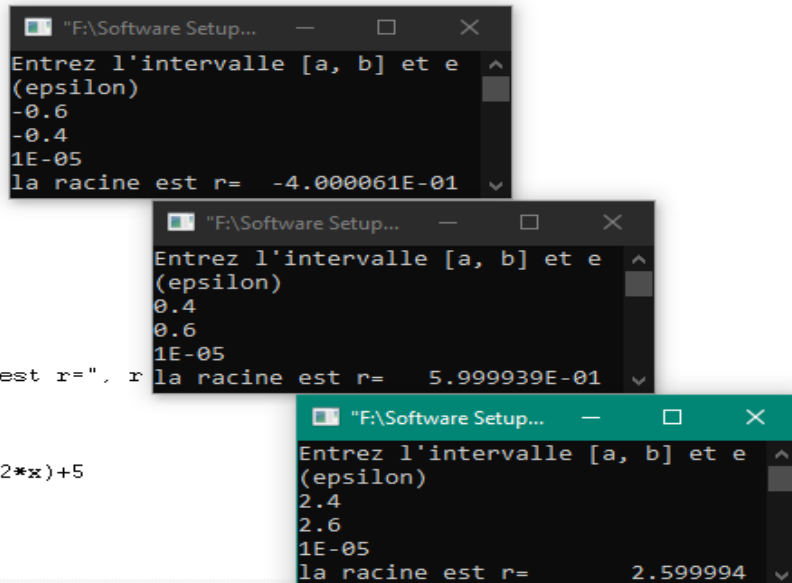


Figure 1.8 :Programme de l'exemple 3 – fonction  $f(x)$

```

Program bissection
implicit none
real:: a, b, c, r, e, g, m
!integer:: i

!----Entrée des données
Write(*,*) "Entrez l'intervalle [a, b] et e (epsilon)"
Read(*,*) a, b, e

!----Boucle
Do
10 c=(b+a)/2
!----Test
if (g(a)*g(c)<0) then
    b=c
    m=(b-a)/2
    if (m<=e) exit
    goto 10
else
    a=c
    m=(b-a)/2
    if (m<=e) exit
    goto 10
endif
enddo
r=(b+a)/2
Write(*,*) "la racine est r=", r

end

Function g(x)
g=(3**x)-(3*x)-2
Return
Endfunction
    
```

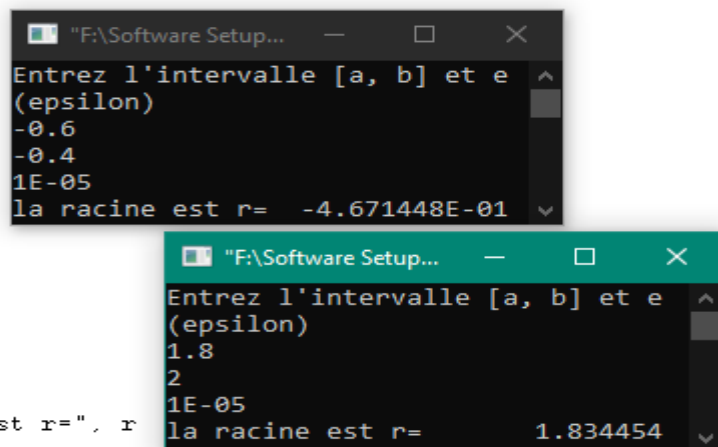


Figure 1.9 :Programme de l'exemple 3 – fonction  $g(x)$

## 1.4 Méthode du point Fixe

### 1.4.1 Définition

Un point fixe d'une fonction  $g(x)$  est une valeur de  $x$  qui reste invariante pour cette fonction  $x = f(x)$ . Pour résoudre des équations non linéaires de la forme  $f(x) = 0$  par cette méthode il suffit de transformer l'équation  $f(x) = 0$  en un problème équivalent  $x = g(x)$ .

A partir d'une certaine valeur initiale  $x_0$  on commence à construire une suite  $x_n$  selon le schéma suivant :

$$x_1 = g(x_0), x_2 = g(x_1), \dots, x_{n+1} = g(x_n) \quad (\text{I.2})$$

Les itérations sont répétées jusqu'à ce que la suite  $x_n$  converge vers la racine avec une certaine précision

$$|x_{n+1} - x_n| < \varepsilon \quad (\text{I.3})$$

A noter que cette méthode ne converge pas toujours. La convergence dépend en général du choix de la fonction  $g(x)$  et du point initial  $x_0$ .

**Exemple :**  $5x^3 - 20x + 3 = 0$  ( $x_0 = 0,5$ ). Dans ce cas on a 3 possibilités pour le choix de  $g$  :

$$g_1(x) = \sqrt[3]{\frac{20x - 3}{5}}; \quad g_2(x) = -\frac{3}{5x^2 - 20} \quad \text{et} \quad g_3(x) = \frac{(5x^3 + 3)}{20}$$

Pour le choix de  $g(x) = g_3(x)$  on a la convergence mais pas pour les deux autres.

### 1.4.2 Théorème

Soit l'équation  $f(x) = 0$  qu'on peut écrire sous la forme  $x = g(x)$ . Alors si :

- i.  $g(x) \in [a, b] \quad \forall x \in [a, b]$
- ii.  $g$  est définie différentiable sur  $[a, b]$
- iii. Il existe un  $k / 0 \leq k < 1$  et  $|g'(x)| < k \quad \forall x \in [a, b]$
- iv. Alors la suite  $\{x_n\}$  ( $x_{n+1} = g(x_n)$ ) converge vers la racine  $r$  quelque soit  $x_0$  et  $r$  est unique.

### 1.4.3 Algorithme

A partir d'une certaine valeur  $x_0$  on effectue les itérations suivantes  $x_{n+1} = g(x_n)$  c.à.d.  $x_1 = g(x_0)$ ,  $x_2 = g(x_1)$ , ...,  $x_{n+1} = g(x_n)$

- i. Choix de  $x_0$
- ii. Effectuer  $x_{n+1} = g(x_n)$
- iii. Si  $|x_{n+1} - x_n| < \varepsilon$  (précision atteinte) donc la convergence est atteinte ; arrêt.
- iv. Si non, retour à l'étape ii.

#### Exemple 1

En utilisant la méthode du **Point Fixe**, trouver la racine de la fonction  $f(x) = 5x^3 - 20x + 3$  avec une précision de  $10^{-4}$  et une valeur initiale de  $x_0 = 0,5$

**Solution :**

$$\text{On détermine } g(x_i) : f(x) = 5x^3 - 20x + 3 \Rightarrow x = \frac{(5x^3 + 3)}{20} \Rightarrow x_{i+1} = g(x_i) = \frac{(5x_i^3 + 3)}{20}$$

Tableau 1.5 : Résultats de calcul appliquant la méthode du point Fixe – exemple 1

$x_{i+1}$	$g(x_i)$	$Abs(\Delta x)$	$\epsilon$
$x_0$	0,5		<b>1,00E-04</b>
$x_1$	0,1813	0,31875	
$x_2$	0,1515	0,02976141	
$x_3$	<b>0,1509</b>	0,00061947	
$x_4$	0,1509	<b>1,0618E-05</b>	

Ayant atteint la précision après 4 itérations, la racine de la fonction  $f(x)$  est  $r \approx x_3 \approx 0,1509$

**Exemple 2**

Avec la méthode du **Point Fixe**, trouvez la racine de la fonction  $f(x) = \cos(x) - xe^x$  avec une précision  $10^{-4}$  et une valeur initiale  $x_0 = 0$ .

**Solution**

On détermine  $g(x_i) : f(x) = \cos(x) - xe^x \Rightarrow x = \frac{\cos(x)}{e^x} \Rightarrow x_{i+1} = g(x_i) = \frac{\cos(x_i)}{e^{x_i}}$

Le tableau 1.6 regroupe les résultats de calculs.

Tableau 1.6 : Résultats de calcul appliquant la méthode du point Fixe – exemple 2

$x_{i+1}$	$g_{x_i}$	$Abs(\Delta x)$	$\epsilon$	$x_{i+1}$	$g_{x_i}$	$Abs(\Delta x)$	$\epsilon$
$x_0$	0		<b>1,00E-04</b>	$x_{23}$	0,5218	9,08E-03	<b>1,00E-04</b>
$x_1$	1	1,00E+00		$x_{24}$	0,5145	7,38E-03	
$x_2$	0,199	8,01E-01		$x_{25}$	0,5204	6,00E-03	
$x_3$	0,804	6,05E-01		$x_{26}$	0,5156	4,88E-03	
$x_4$	0,311	4,93E-01		$x_{27}$	0,5195	3,96E-03	
$x_5$	0,698	3,87E-01		$x_{28}$	0,5163	3,22E-03	
$x_6$	0,381	3,16E-01		$x_{29}$	0,5189	2,62E-03	
$x_7$	0,634	2,53E-01		$x_{30}$	0,5168	2,13E-03	
$x_8$	0,427	2,06E-01		$x_{31}$	0,5185	1,73E-03	
$x_9$	0,593	1,66E-01		$x_{32}$	0,5171	1,40E-03	
$x_{10}$	0,458	1,36E-01		$x_{33}$	0,5183	1,14E-03	
$x_{11}$	0,567	1,09E-01		$x_{34}$	0,5173	9,27E-04	
$x_{12}$	0,478	8,92E-02		$x_{35}$	0,5181	7,54E-04	
$x_{13}$	0,550	7,23E-02		$x_{36}$	0,5175	6,12E-04	
$x_{14}$	0,492	5,89E-02		$x_{37}$	0,5180	4,98E-04	
$x_{15}$	0,539	4,77E-02		$x_{38}$	0,5176	4,04E-04	
$x_{16}$	0,500	3,88E-02		$x_{39}$	0,5179	3,29E-04	
$x_{17}$	0,532	3,15E-02		$x_{40}$	0,5176	2,67E-04	
$x_{18}$	0,506	2,56E-02		$x_{41}$	0,5179	2,17E-04	
$x_{19}$	0,527	2,08E-02		$x_{42}$	0,5177	1,76E-04	
$x_{20}$	0,510	1,69E-02		$x_{43}$	0,5178	1,43E-04	
$x_{21}$	0,524	1,37E-02		$x_{44}$	<b>0,5177</b>	1,17E-04	
$x_{22}$	0,513	1,12E-02	$x_{45}$	0,5178	<b>9,47E-05</b>		

On atteint la précision après 45 itérations. On déduit que la racine de la fonction  $f(x) = \cos(x) - xe^x$  est :  $r \approx x_{44} \approx 0,5177$

**Exemple 3**

Avec la méthode du **Point Fixe**, calculez la racine de la fonction  $f(x) = 3x + \sin(x) - e^x$  avec une erreur relative de 0,01 et une valeur initiale  $x_0 = 1$

**Solution**

On détermine  $g(x_i) : f(x) = 3x + \sin(x) - e^x \Rightarrow x = \frac{e^x - \sin(x)}{3} \Rightarrow x_{i+1} = g(x_i) = \frac{e^{x_i} - \sin(x_i)}{3}$

Le tableau 1.7 regroupe les résultats de calculs. La précision est atteinte après cinq (05) itérations.

On déduit que la racine de la fonction  $f(x) = 3x + \sin(x) - e^x$  est  $r \approx x_4 \approx 0,3625$

*Tableau 1.7 : Résultats de calcul appliquant la méthode du point Fixe – exemple 3*

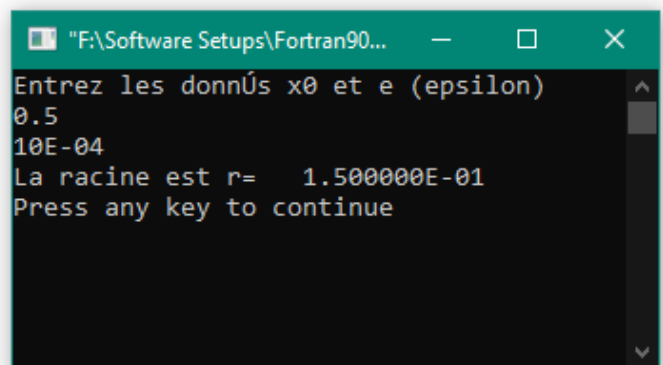
$x_{i+1}$	$g_{x_i}$	$Abs(\Delta x)$	$\epsilon$
$x_0$	1		<b>1,00E-02</b>
$x_1$	0,63	3,74E-01	
$x_2$	0,43	1,98E-01	
$x_3$	0,37	5,49E-02	
$x_4$	<b>0,36</b>	1,05E-02	
$x_5$	0,36	<b>1,78E-03</b>	

**1.4.4 Programme de la « méthode du Point Fixe »**

**i. Exemple 1**

```

Program Point_Fixe
Implicit none
real:: r, e, g
real, dimension(1:1000000):: x
integer:: i
!----Entrée des données
Write(*,*) "Entrez les donnés x0 et e (epsilon)"
Read(*,*) x(0), e
!----Boucle
i=0
Do
i=i+1
x(i+1)=g(x(i))
!----Test
if (Abs(x(i+1)-x(i))<e) exit
Enddo
!----Affichage de la racine
r=x(i)
Write(*,*) "La racine est r=", r
End
!----La Fonction
Function g(x)
g=((5*x**3)+3)/20
Return
Endfunction
    
```



*Figure 1.10 : Programme Méthode du POINT FIXE appliqué à l'exemple 1*

**ii. Exemple 2**

```

Program Point_Fixe
Implicit none
real:: r, e, g
real, dimension(1:1000000):: x
integer:: i
!----Entrée des données
Write(*,*) "Entrez les donnés x0 et e (epsilon)"
Read(*,*) x(0), e
!----Boucle
i=0
Do
i=i+1
x(i+1)=g(x(i))
!----Test
if (Abs(x(i+1)-x(i))<e) exit
Enddo
!----Affichage de la racine
r=x(i)
Write(*,*) "La racine est r=", r
End
!----La Fonction
Function g(x)
g=(cos(x))/(exp(x))
Return
Endfunction

```

```

F:\Software Setups\Fortran...
Entrez les donnés x0 et e (epsilon)
0
0.0001
La racine est r= 5.177051E-01
Press any key to continue

```

Figure 1.11 :Programme Méthode du POINT FIXE appliqué à l'exemple 2

### iii. Exemple 3

```

Program Point_Fixe
Implicit none
real:: r, e, g
real, dimension(1:1000000):: x
integer:: i
!----Entrée des données
Write(*,*) "Entrez les donnés x0 et e (epsilon)"
Read(*,*) x(0), e
!----Boucle
i=0
Do
i=i+1
x(i+1)=g(x(i))
!----Test
if (Abs(x(i+1)-x(i))<e) exit
Enddo
!----Affichage de la racine
r=x(i)
Write(*,*) "La racine est r=", r
End
!----La Fonction
Function g(x)
g=((exp(x))- sin(x))/3
Return
Endfunction

```

```

F:\Software Setups\Fortran...
Entrez les donnés x0 et e (epsilon)
1
1E-02
La racine est r= 3.561392E-01
Press any key to continue

```

Figure 1.12 :Programme Méthode du POINT FIXE appliqué à l'exemple 3

## 1.5 Méthode de Newton – Raphson

### 1.5.1 Définition

Cette méthode est basée sur le développement de Taylor au voisinage d'un point (négligeant les termes d'ordre supérieur) tel que :

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \dots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n + \dots \quad (I.4)$$

$f(r) = 0$  ; soit  $r = x + h$

$$f(x + h) = f(x) + h \times f'(x) + h^2 \times f''(x)/2! \quad (\text{I.5})$$

À l'ordre 1, l'équation I.5 devient :

$$f(x + h) = f(x) + h \times f'(x) + O(h^2) \quad (\text{I.6})$$

Dans le cas où  $x + h = r$ , on obtient :

$$f(r) = f(x) + h \times f'(x) + O(h^2) = 0 \Rightarrow h = -\frac{f(x)}{f'(x)} \quad (\text{I.7})$$

On construit une suite sur cette définition c.à.d. on s'approche de  $r$  à partir d'un certain point initial  $x_0$  en ajoutant à chaque fois un accroissement  $h = -f(x)/f'(x)$  comme suit :

$$h = \frac{f(x_0)}{f'(x_0)} ; x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} ; x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} ; \dots ; x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (\text{I.8})$$

On est d'autant plus proche de  $r$  si le rapport de  $f$  et  $f'$  est de l'ordre zéro

$$|r - x| = |f(x)/f'(x)| \approx O(0) \quad (\text{I.9})$$

On s'arrête dès que la précision est atteinte c.à.d.  $|x_{n+1} - x_n| < \varepsilon$ .

### 1.5.2 L'algorithme

- i. Choix de  $x_0$
- ii. Calcul de  $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$
- iii. Test de précision  $|x_{i+1} - x_i| < \varepsilon$ . Si oui, on s'arrête.
- iv. Si non, retour l'étape ii.

#### Exemple

En utilisant la méthode de **Newton**, résoudre l'équation suivante :  $f(x) = (x - 1)^2 = 0$  avec  $x_0 = 0$  et une précision de  $\varepsilon = 10^{-10}$

#### Solution

On utilise le programme **FORTRAN** qui réduit considérablement le temps de résolution. La figure 1.13 montre le programme avec le résultat de compilation.

### 1.5.3 Programme de la « Méthode de Newton – Raphson »

```

Program Newton
Implicit none
doubleprecision:: e, r
real:: f, fp
doubleprecision, dimension(1:1000000):: x
integer:: i

Write(*,*) "Entrez les donnés x0 et e (epsilon)"
Read(*,*) x(0), e

i=0
Do
i=i+1
x(i+1)=x(i)-(f(x(i))/fp(x(i)))

!----Test de précision
if (Abs(x(i+1)-x(i))<e) exit

Enddo

r=x(i)
Write(*,*) "La racine est r=", r
End

!----La Fonction
Function f(x)
doubleprecision:: x
f=(x-1)**2
Return
Endfunction
!----La Dérivée
Function fp(x)
doubleprecision:: x
fp=2*(x-1)
Return
Endfunction

```

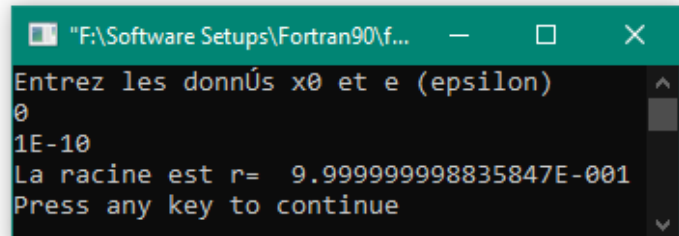


Figure 1.13 : Programme de la méthode de Newton – Raphson

## 1.6 Racines Multiples

### 1.6.1 Définition

Une fonction  $f$ , dont la racine  $r$  est dite de multiplicité  $m$ , est de la forme :

$$f(x) = (x - r)^m h(x) \text{ avec } h(r) \neq 0 \quad (\text{I.10})$$

### 1.6.2 Théorème

Une racine  $r$  est de multiplicité  $m$  si  $f'(r) = f''(r) = \dots = f^{(m)}(r) = 0$  et  $f^{(m+1)}(r) \neq 0$

## 1.7 Cas de fonction Polynomiale

Un polynôme de degré  $n$  général est de la forme  $p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0$  où  $\{a_i\} \in \mathbb{C}$  ou  $\mathbb{R}$

### 1.7.1 Evaluation de Polynôme (Schéma de HORNER)

L'évaluation direct du polynôme nécessite  $n(n+1)/2$  multiplications et  $n$  additions. Pour diminuer la complexité, Horner proposa de factoriser  $p(x)$  sous la forme :

$$p(x) = \left( \left( \dots \left( (a_n x + a_{n-1}) x + a_{n-2} \right) x + \dots a_1 \right) x + a_0 \right) \quad (\text{I.11})$$

On obtient les relations de récurrence suivante :

$$b_0 = a_n; b_1 = b_0x + a_{n-1}; \dots; b_k = b_{k-1}x + a_{n-k}; \dots; b_n = b_{n-1}x + a_0 \quad (\text{I.12})$$

Dans ce cas la valeur du polynôme  $p_n(x)$  au point  $x_0$  est  $p_n(x_0) = b_n = b_{n-1}x_0 + a_0$  qu'on obtient après avoir calculé récursivement  $b_1, b_2, \dots, b_{n-1}$ .

### 1.7.1.1 Programme du « Schéma de Horner »

```

Program schema_de_Horner
implicit none
integer :: n,k
doubleprecision :: x0
integer ,parameter :: ndim=3
doubleprecision , dimension(0:ndim):: a,b

!---L'ordre du polynôme
print*,'donner l''ordre du polynome p(x)=a0+a1*x+...an*x^n'
read(*,*) n

!---Lecture des coefficient a(i)
do i=0,n
print*,'donner la valeur du coefficient' i
read(*,*) a(i)
enddo

!---Calcul de P(x0)
print*,'donner la valeur x0 à laquelle vous voulez calculer P(x)'
read(*,*) x0
b(0)=a(n)
do k=1,n
b(k)=b(k-1)*x0+a(n-k)
enddo

!---Affichage du résultat
write(*,*) 'la valeur du plynome p(x) au point x0 est',b(n)
end

```

Figure 1.14 :Programme du Schéma de HORNER

## 1.7.2 Division de Polynôme

Soit  $b(x)$  un polynôme de degré  $p < n$  en divisant  $p(x)$  par  $b(x)$  on aura en générale :

$$p(x) = q(x)b(x) + R(x) \quad (\text{I.13})$$

Où  $q(x)$  est le quotient et  $R(x)$  est le reste (degré de  $R <$  degré de  $q$  évidemment).

Cas particulier où  $b(x) = (x - r) \Rightarrow p(x) = (x - r)q(x) + R(x) \quad (\text{I.14})$

- Si  $\text{degré}[p(x)] = n$  alors  $\text{degré}[q(x)] = (n - 1)$  et  $\text{degré}[R(x)] = 0$  Donc  $R(x)$  est constante par conséquent.
- Si  $r$  est une racine de  $p$  alors  $(x - r)$  est un diviseur de  $p(x)$  et  $R(x) = 0$

De plus  $p'(x) = (x - r)q'(x) + q(x) \Rightarrow p'(r) = q(r)$

## 1.7.3 Séparation et Calcul du Nombre des Racines

### 1.7.3.1 Règle de DESCARTES

Soit  $\{a_i; i = 0 \dots n\}$  la suite des coefficients de  $p(x)$  supposés réels (certains peuvent être nuls) et soit  $k$  le nombre de changement de signe de cette suite alors le nombre  $p$  de racines réelles positives vérifie  $p \leq k$  et  $k - p = \text{pair}$ . Le nombre des racines négatives est obtenu en appliquant cette règle au polynôme  $p(-x)$ .

### Exemple



$p(x) = x^6 - x - 1$ ;  $k = 1 \Rightarrow p \leq 1 \Rightarrow p = 0$ ; 1 et  $(k - p) = \text{pair} \Rightarrow \mathbf{p} = \mathbf{1}$  (Une racine positive)

$p(-x) = x^6 + x - 1$ ;  $k = 1$  et  $\mathbf{p} = \mathbf{1}$  (Une racine négative)

### 1.7.3.2 Théorème de STURM

A partir du polynôme  $p_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0$ , on construit la suite de polynômes suivants :

$$\left\{ \begin{array}{l} p_0(x) = p(x); \\ p_1(x) = p'(x); \\ p_2(x) = -R_1(x); \\ p_3(x) = -R_2(x); \\ \vdots \\ p_k(x) = -R_{k-1}(x); \end{array} \right\} \text{ où } \left\{ \begin{array}{l} R_1(x) = \text{reste de la division de } \frac{p_0(x)}{p_1(x)} \\ R_2(x) = \text{reste de la division de } \frac{p_1(x)}{p_2(x)} \\ \vdots \\ R_{k-1}(x) = \text{reste de la division de } \frac{p_{k-2}(x)}{p_{k-1}(x)} \end{array} \right. \quad (I.15)$$

On s'arrête à  $p_m(x)$  quand  $R_m$  devient constant.

La suite  $\{p_0(x); p_1(x); p_2(x); \dots; p_m(x)\}$  est appelée « Suite de STURM ».

Le nombre des racines réelles de  $p(x)$  dans l'intervalle  $[a, b]$  est égale à  $|w(b) - w(a)|$  où  $w(x)$  est le nombre de changement de signe de la suite  $\{p(x)\}$ .

#### 1.7.3.2.1 Programme du « Théorème de STURM »

```

Program STURM
implicit none
doubleprecision :: x1,x2
integer ,parameter :: ndim=300,poly=10
doubleprecision , dimension(0:ndim,0:poly):: b1,b2,a
integer :: i,n,k,p,m,j,ncsa,ncsb
write(*,*) "donner l'ordre du polynome p(x)=a0+a1*x+...an*x^n"
read(*,*) n,m
write(*,*) "donner la suite de STURM p0(x), p1(x)...pn(x)"
p=0
5 p=p+1
do i=0,p
do j=0,m
write(*,*) "donner la valeur du coefficient", i," du polynome",j
read(*,*) a(i,j)
enddo
enddo
if(p< m) goto 5
! calcul de la valeur de la suite de sturm pi(a) ,pi(b) au borne de l'intervalle'
do j=0,m
b1(0,j)=a(n,j)
b2(0,j)=a(n,j)
do k=1,n
b1(k,j)=b1(k-1,j)*x1+a(n-k,j)
b2(k,j)=b2(k-1,j)*x2+a(n-k,j)
enddo
enddo
! calcul du nombre de changement de signe w(a) et w(b) puis |w(a)-w(b)|
ncsa=0
ncsb=0
do j=0,m-1
if(b1(n,j)*b1(n,j+1)<0) ncsa=ncsa+1
if(b2(n,j)*b2(n,j+1)<0) ncsb=ncsb+1
enddo
write(*,*) ncsa, ncsb
End

```

Figure 1.15 :Programme FORTRAN du Théorème de STURM

**Chapitre 2**  
**SYSTEME D'EQUATIONS**  
**LINEAIRES**

---

**Chapitre 2 SYSTEME D'EQUATIONS LINEAIRES**

**2.1 Introduction**

Un système linéaire à n variables avec n équations est un ensemble d'équations de la forme suivante :

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \dots a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 \dots a_{2n}x_n = b_2 \\ \dots \\ \dots \\ a_{p1}x_1 + a_{p2}x_2 + a_{p3}x_3 \dots a_{pn}x_n = b_p \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 \dots a_{nn}x_n = b_n \end{cases} \tag{II. 1}$$

Les  $a_{ij}$  et  $b_i (i, j = 1 \dots n)$  sont des paramètres numériques donnés et les  $x_i (i = 1 \dots n)$  sont les inconnues. La solution du système consiste à trouver l'ensemble des valeurs  $x_i \ i = 1 \dots n$  qui vérifient ces équations en même temps.

On peut l'écrire sous forme matricielle :  $AX=B$

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{p1} & a_{p2} & a_{p3} & \dots & a_{pn} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_p \\ \vdots \\ b_n \end{pmatrix} \tag{II. 2}$$

Ce type de système d'équations linéaires joue un rôle important dans divers domaines de physique et électronique. On s'intéresse particulièrement au système de Cramer c.à.d. ceux qui ont  $det(A) \neq 0$  dans ce cas la solution du système est unique.

Dans le cas où le nombre d'équations et par conséquent d'inconnus est élevés les méthodes de résolution d'algèbre usuelles sont inapplicables et l'utilisation des méthodes numériques s'avère inévitable.

**2.2 Méthodes d'Elimination de Gauss (Triangularisation)**

**2.2.1 Définition**

Cette méthode est basée sur l'élimination des variables une à une jusqu'à rendre le système ou la matrice A triangulaire. Premièrement, on commence par définir une matrice triangulaire.

**Définition :** Une matrice est dite triangulaire si elle est de la forme suivante :

Matrice triangulaire inférieure avec $a_{ij} = 0 \text{ si } i < j$	Matrice triangulaire supérieure avec $a_{ij} = 0 \text{ si } i > j$
$\begin{pmatrix} a'_{11} & 0 & 0 & 0 & \dots & 0 \\ a'_{21} & a'_{22} & 0 & 0 & \dots & 0 \\ a'_{31} & a'_{32} & a'_{33} & \dots & \dots & 0 \\ \dots & \dots & \dots & a'_{pp} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \vdots \\ a'_{n1} & a'_{n2} & a'_{n3} & \dots & \dots & a'_{nn} \end{pmatrix}$	$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ 0 & a'_{22} & a'_{23} & \dots & a'_{2n} \\ 0 & 0 & a'_{33} & \dots & a'_{3n} \\ 0 & 0 & \dots & a'_{pp} & a'_{pn} \\ 0 & 0 & 0 & \dots & \vdots \\ 0 & 0 & 0 & \dots & a'_{nn} \end{pmatrix}$

Pour un système à  $n$  équations, on a besoin en générale de  $(n - 1)$  étapes (éliminations) et à chaque étape on aura un nouveau système dont les coefficients  $a_{ij}$  et  $b_i$  sont différent de ceux du système précédent.

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1 & \text{equ. (1)} \\ 0 + a'_{22}x_2 + a'_{23}x_3 + \dots + a'_{2n}x_n = b'_2 & \text{equ. (2)} \\ \dots \dots \dots \dots \dots \\ 0 + 0 + \dots \dots + a'_{pp}x_p + \dots + a'_{pn}x_n = b'_p & \text{equ. (p)} \\ \dots \dots \dots \dots \dots \\ 0 + 0 + \dots \dots + \dots \dots + 0 + a'_{nn}x_n = b'_n & \text{equ. (n)} \end{cases} \quad \text{(II. 3)}$$

En notation  
matricielle

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ 0 & a'_{22} & a'_{23} & \dots & a'_{2n} \\ 0 & 0 & \dots & \dots & a'_{3n} \\ 0 & \dots & a'_{pp} & \dots & a'_{pn} \\ 0 & 0 & \dots & \dots & \vdots \\ 0 & 0 & 0 & \dots & a'_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b'_2 \\ \vdots \\ b'_p \\ \vdots \\ b'_n \end{pmatrix} \quad \text{(II. 4)}$$

### 2.2.2 L'algorithme

Pour calculer la solution on commence par la dernière équation d'où on calcule  $x_n = \frac{b'_n}{a'_{nn}}$  puis on remplace dans l'équation précédente  $(n - 1)$  pour avoir  $x_{n-1}$  et ainsi de suite jusqu'à la première équation pour obtenir  $x_1$  (en remontée)

#### Exemple de Triangularisation

Pour illustrer cette méthode effectuant la triangularisation, on applique la méthode d'élimination de Gauss à un système de trois équations.

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 & \text{equ. (1)} \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 & \text{equ. (2)} \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 & \text{equ. (3)} \end{cases} \quad \text{En notation matricielle} \Rightarrow \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

**Etapes 1 :** Elimination de la variable  $x_1$  des équations (2) et (3). Pour cela :

a- On multiplie l'équation (1) par  $\left(-\frac{a_{21}}{a_{11}}\right)$  on aura une nouvelle équation (1') :

$$(1) \times \left(-\frac{a_{21}}{a_{11}}\right) \Rightarrow -a_{21}x_1 - \frac{a_{21}}{a_{11}}a_{12}x_2 - \frac{a_{21}}{a_{11}}a_{13}x_3 = -\frac{a_{21}}{a_{11}}b_1 \quad \dots (1')$$

b- On ajoute l'équation obtenue (1') à l'équation (2) ((1') + (2)) on aura :

$$(1') + (2) \Rightarrow 0 \cdot x_1 + \left(a_{22} - \frac{a_{21}}{a_{11}}a_{12}\right)x_2 + \left(a_{23} - \frac{a_{21}}{a_{11}}a_{13}\right)x_3 = \left(b_2 - \frac{a_{21}}{a_{11}}b_1\right)$$

De la forme  $0 \cdot x_1 + a'_{22}x_2 + a'_{23}x_3 = b'_2$

Puis de la même manière on élimine  $x_1$  de l'équation (3)

c- On multiplie l'équation (1) par  $\left(-\frac{a_{31}}{a_{11}}\right)$  on aura une nouvelle équation (1'') :

$$(1) \times \left( -\frac{a_{31}}{a_{11}} \right) \Rightarrow -a_{31}x_1 - \frac{a_{31}}{a_{11}}a_{12}x_2 - \frac{a_{31}}{a_{11}}a_{13}x_3 = -\frac{a_{31}}{a_{11}}b_1 \quad \dots (1'')$$

d- On ajoute l'équation obtenue (1'') à l'équation (3) ((1'') + (3)) on aura :

$$(1'') + (3) \Rightarrow 0.x_1 + \left( a_{32} - \frac{a_{31}}{a_{11}}a_{12} \right)x_2 + \left( a_{33} - \frac{a_{31}}{a_{11}}a_{13} \right)x_3 = \left( b_3 - \frac{a_{31}}{a_{11}}b_1 \right)$$

De la forme 
$$0.x_1 + a'_{32}x_2 + a'_{33}x_3 = b'_2$$

Le système après l'étape 1 s'écrit donc :

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 & \text{equ. (1)} \\ 0.x_1 + a'_{22}x_2 + a'_{23}x_3 = b'_2 & \text{equ. (2')} \\ 0.x_1 + a'_{32}x_2 + a'_{33}x_3 = b'_3 & \text{equ. (3')} \end{cases}$$

Avec :

$$\begin{aligned} a'_{22} &= \left( a_{22} - \frac{a_{21}}{a_{11}}a_{12} \right); \quad a'_{23} = \left( a_{23} - \frac{a_{21}}{a_{11}}a_{13} \right); \quad a'_{32} = \left( a_{32} - \frac{a_{31}}{a_{11}}a_{12} \right); \quad a'_{33} \\ &= \left( a_{33} - \frac{a_{31}}{a_{11}}a_{13} \right); \\ b'_2 &= \left( b_2 - \frac{a_{21}}{a_{11}}b_1 \right); \quad b'_3 = \left( b_3 - \frac{a_{31}}{a_{11}}b_1 \right) \end{aligned}$$

**Etape 2** Elimination de la variable  $x_2$  de l'équation (3') du nouveau système

a- On procède de la même manière, on multiplie l'équation(2') par  $\left( -\frac{a'_{32}}{a'_{22}} \right)$  on obtient l'équation (2''):

$$(2') \times \left( -\frac{a'_{32}}{a'_{22}} \right) \Rightarrow 0.x_1 - a'_{32}x_2 - \frac{a'_{32}}{a'_{22}}a'_{23}x_3 = -\frac{a'_{32}}{a'_{22}}b'_2 \quad \dots (2'')$$

b- On ajoute l'équation obtenue (2'') à l'équation (3')((2'') + (3')) on aura la nouvelle équation :

$$(2'') + (3') \Rightarrow 0.x_1 + 0.x_2 + \left( a'_{33} - \frac{a'_{32}}{a'_{22}}a'_{13} \right)x_3 = \left( b'_3 - \frac{a'_{32}}{a'_{22}}b'_2 \right)$$

De la forme 
$$0.x_1 + 0.x_2 + a''_{33}x_3 = b''_2$$

Le système après l'étape 2 s'écrit donc :

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 & \text{equ. (1)} \\ 0.x_1 + a'_{22}x_2 + a'_{23}x_3 = b'_2 & \text{equ. (2')} \\ 0.x_1 + 0.x_2 + a''_{33}x_3 = b''_2 & \text{equ. (3'')} \end{cases}$$

Avec :  $a''_{33} = \left( a'_{33} - \frac{a'_{32}}{a'_{22}}a'_{13} \right)$  et  $b''_2 = \left( b'_3 - \frac{a'_{32}}{a'_{22}}b'_2 \right)$

Donc à l'étape 2 le système est devenu triangulaire.

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & 0 & a''_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b'_2 \\ b''_3 \end{pmatrix}$$

On le résout par remontée de l'équation (3'') comme décrit dans les étapes suivantes :

- (a) On calcule  $x_3 = \frac{b''_3}{a''_{33}}$
- (b) On remplace  $x_3$  dans l'équation (2') pour obtenir  $x_2 = \frac{b'_2 - a'_{23}x_3}{a'_{22}}$
- (c) Finalement, on extrait de l'équation (1)  $x_1 = \frac{b_1 - a_{12}x_2 - a_{13}x_3}{a_{11}}$
- (d) En généralisant la formule, on obtient :

$$x_i = \frac{b_i - \sum_{i=1}^n a_{i(i+1)}x_{i+1}}{a_{ii}} \tag{II. 5}$$

**Exemple d'un système (3 × 3)**

$$\begin{cases} 2x_1 + x_2 - 4x_3 = 8 & \dots (1) \\ 3x_1 + 3x_2 - 5x_3 = 14 & \dots (2) \\ 4x_1 + 5x_2 - 2x_3 = 16 & \dots (3) \end{cases} \quad \begin{array}{l} \text{En notation} \\ \text{matricielle} \Rightarrow \end{array} \quad \begin{pmatrix} 2 & 1 & -4 \\ 3 & 3 & -5 \\ 4 & 5 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 8 \\ 14 \\ 16 \end{pmatrix}$$

**Étape 1** : On calcule les coefficients  $a'_{ij}$  du nouveau système où on élimine  $x_1$  des équations (2) et (3)

$$a'_{22} = \left( a_{22} - \frac{a_{21}}{a_{11}} a_{12} \right) = \left( 3 - \frac{3}{2} \cdot 1 \right) = \frac{3}{2} \quad a'_{23} = \left( a_{23} - \frac{a_{21}}{a_{11}} a_{13} \right) = \left( -5 - \frac{3}{2}(-4) \right) = 1$$

$$a'_{32} = \left( a_{32} - \frac{a_{31}}{a_{11}} a_{12} \right) = \left( 5 - \frac{4}{2} \cdot 1 \right) = 3 \quad a'_{33} = \left( a_{33} - \frac{a_{31}}{a_{11}} a_{13} \right) = \left( -2 - \frac{4}{2}(-4) \right) = 6$$

$$b'_2 = \left( b_2 - \frac{a_{21}}{a_{11}} b_1 \right) = \left( 14 - \frac{3}{2} \cdot 8 \right) = 2 \quad b'_3 = \left( b_3 - \frac{a_{31}}{a_{11}} b_1 \right) = \left( 16 - \frac{4}{2} \cdot 8 \right) = 0$$

Donc au terme de **l'étape 1**, nous avons éliminé la variable  $x_1$  des équations (2) et (3). Le système devient alors :

$$\begin{cases} 2x_1 + x_2 - 4x_3 = 8 & \dots (1) \\ 0 + \frac{3}{2}x_2 + x_3 = 2 & \dots (2') \\ 0 + 3x_2 + 6x_3 = 0 & \dots (3') \end{cases}$$

**Étape 2** : On calcule les coefficients  $a''_{ij}$  du nouveau système où on élimine  $x_2$  de l'équation (3') :

$$a''_{33} = \left( a'_{33} - \frac{a'_{32}}{a'_{22}} a'_{23} \right) = \left( 6 - \frac{3}{\frac{3}{2}} \cdot 1 \right) = 4 \quad b''_3 = \left( b'_3 - \frac{a'_{32}}{a'_{22}} b'_2 \right) = \left( 0 - \frac{3}{\frac{3}{2}} \cdot 2 \right) = -4$$

Suite à **l'étape 2** le nouveau système prend la forme triangulaire recherchée :

$$\begin{cases} 2x_1 + x_2 - 4x_3 = 8 & \dots (1) \\ 0 + \frac{3}{2}x_2 + x_3 = 2 & \dots (2') \\ 0 + 0 + 4x_3 = -4 & \dots (3'') \end{cases}$$

Désormais, on peut aisément calculer les variables en remontant de la variable  $x_3$  tel que :

- De l'équation (3'')  $\Rightarrow x_3 = -1$
- On remplace  $x_3$  dans (2')  $\Rightarrow x_2 = 2$
- On remplace  $x_3$  et  $x_2$  dans (1)  $\Rightarrow x_1 = 1$   $\Rightarrow$  Enfin la solution du système est  $\begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}$

**2.2.3 Généralisation de la Triangularisation pour un système ( $N \times N$ )**

On a eu besoin de 2 étapes pour rendre triangulaire le système à ( $3 \times 3$ ). Les coefficients de chaque étape s'obtiennent à partir des coefficients de l'étape précédente. En générale pour un système ( $N \times N$ ) c.à.d.  $N$  équations à  $N$  inconnues, on aura besoin en générale de ( $N - 1$ ) étapes pour triangulariser le système. À une étape ( $p$ ) quelconque, les coefficients s'écrivent :

$$a_{ij}^{(p)} = \left( a_{ij}^{(p-1)} - \frac{a_{ip}^{(p-1)}}{a_{pp}^{(p-1)}} a_{pj}^{(p-1)} \right) \text{ et } b_i^{(p)} = \left( b_i^{(p-1)} - \frac{a_{ip}^{(p-1)}}{a_{pp}^{(p-1)}} b_p^{(p-1)} \right) \quad (\text{II. 6})$$

- Les  $a_{ij}^{(p)}$  et  $b_i^{(p)}$  sont les coefficients à l'étape ( $p$ )
- Les  $a_{ij}^{(p-1)}$  et  $b_i^{(p-1)}$  sont les coefficients à l'étape ( $p-1$ )
- Les  $a_{ij}^{(0)}$  et  $b_i^{(0)}$  sont les coefficients du système initiale  $a_{ij}$  et  $b_i$
- Les  $a_{ij}^{(p)} = 0$  pour  $i \leq p$  ou  $j \leq p$  Donc il est inutile de calculer ces éléments

Dans l'exemple ( $3 \times 3$ ) qu'on a vu précédemment, les  $a'_{ij}$  et  $b'_i$  sont les coefficients à l'étape (1) ( $a'_{ij} = a_{ij}^{(1)}$  et  $b'_i = b_i^{(1)}$ ) alors que les  $a''_{ij}$  et  $b''_i$  sont ceux de l'étape (2) ( $a''_{ij} = a_{ij}^{(2)}$  et  $b''_i = b_i^{(2)}$ ).

A une étape ( $p$ ) quelconque le système ( $N \times N$ ) a la forme générale suivante :

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1 \\ 0 + a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + \dots + a_{2n}^{(1)}x_n = b_2^{(1)} \\ 0 + 0 + a_{33}^{(2)}x_3 + \dots + a_{3n}^{(2)}x_n = b_3^{(2)} \\ \dots \\ 0 + \dots + 0 + \dots + a_{pp}^{(p-1)}x_p + a_{pn}^{(p-1)}x_n = b_p^{(p-1)} \\ 0 + \dots + 0 + \dots + a_{(p-1)n}^{(p)}x_p + \dots + a_{nn}^{(p)}x_n = b_n^{(p)} \end{cases} \quad (\text{II. 7})$$

C.à.d. on a éliminé les ( $p - 1$ ) première variables  $x_i (i = 1, \dots, p - 1)$  de toutes les équations du système.

A l'étape ( $n - 1$ ) finale on aura éliminé toute les variables  $x_i (i = 1, \dots, n - 1)$  de toutes les équations du système qui devient alors triangulaire.

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1 \\ 0 + a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + \dots + a_{2n}^{(1)}x_n = b_2^{(1)} \\ 0 + 0 + a_{33}^{(2)}x_3 + \dots + a_{3n}^{(2)}x_n = b_3^{(2)} \\ \dots \\ 0 + \dots + 0 + \dots + a_{pp}^{(p-1)}x_p + a_{pn}^{(p-1)}x_n = b_p^{(p-1)} \\ 0 + \dots + 0 + \dots + 0 + \dots + a_{nn}^{(n-1)}x_n = b_n^{(n-1)} \end{cases} \quad (\text{II. 8})$$

En notation matricielle :

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & \dots & a_{1n} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & \dots & a_{2n}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & \dots & a_{3n}^{(2)} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{pp}^{(p-1)} & \dots & a_{pn}^{(p-1)} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \dots & \dots & 0a_{nn}^{(n-1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_p \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2^{(1)} \\ b_3^{(2)} \\ \vdots \\ b_p^{(p-1)} \\ \vdots \\ b_n^{(n-1)} \end{pmatrix}$$

**N.B :** Les coefficients  $a_{ii}^{(p)}$  s'appellent les pivots et doivent être non nuls.

**Exemple 1**

Le système suivant  $\begin{cases} x_1 + x_2 + x_3 = 6 \\ x_1 + 3x_2 + 2x_3 = 13 \\ x_1 + x_2 - x_3 = 0 \end{cases} \Rightarrow \begin{pmatrix} 1 & 1 & 1 \\ 1 & 3 & 2 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 13 \\ 0 \end{pmatrix}$

On a un système  $3 \times 3$  c.à.d. on aura besoin en générale de  $(n - 1) = 2$  étapes pour rendre le système triangulaire en utilisant les relations (II. 6) :

$$a_{ij}^{(p)} = \left( a_{ij}^{(p-1)} - \frac{a_{ip}^{(p-1)}}{a_{pp}^{(p-1)}} a_{pj}^{(p-1)} \right) \text{ et } b_i^{(p)} = \left( b_i^{(p-1)} - \frac{a_{ip}^{(p-1)}}{a_{pp}^{(p-1)}} b_p^{(p-1)} \right)$$

Il faut se rappeler que les coefficients  $a_{ij}^{(p)}$  tel que  $i \leq p$  ou  $j \leq p$  sont nuls donc ce n'est pas la peine de les calculer.

**Etape 1 :** on calcule les coefficients du nouveau système  $a_{ij}^{(1)}$  et  $b_i^{(1)}$  (de l'étape 1)  $p = 1$  :

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ 0. x_1 + a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 &= b_2^{(1)} \\ 0. x_1 + a_{32}^{(1)}x_2 + a_{33}^{(1)}x_3 &= b_3^{(1)} \end{aligned}$$

Les coefficients de la 1<sup>ère</sup> équation ne changent pas  $a_{11}^{(0)} = a_{11}$ ,  $a_{12}^{(0)} = a_{12}$ ,  $a_{13}^{(0)} = a_{13}$ ,  $b_1^{(0)} = b_1$

**Etape 2 :** Puis on calcule les coefficients  $a_{ij}^{(2)}$  et  $b_i^{(2)}$  (de l'étape 2)  $p = 2$  :

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ 0. x_1 + a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 &= b_2^{(1)} \\ 0. x_1 + 0. x_2 + a_{33}^{(2)}x_3 &= b_3^{(2)} \end{aligned}$$

De (II.6), on a les coefficients du système à l'étape 1 :

$$a_{ij}^{(1)} = \left( a_{ij}^{(0)} - \frac{a_{i1}^{(0)}}{a_{11}^{(0)}} a_{1j}^{(0)} \right) \quad \text{Et} \quad b_i^{(1)} = \left( b_i^{(0)} - \frac{a_{i1}^{(0)}}{a_{11}^{(0)}} b_1^{(0)} \right)$$

On commence le calcul de chaque élément avec  $i > p$  et  $j > p$  :

- $i = 2$  et  $j = 2 \Rightarrow a_{22}^{(1)} = \left( a_{22}^{(0)} - \frac{a_{21}^{(0)}}{a_{11}^{(0)}} a_{12}^{(0)} \right) = \left( 3 - \frac{1}{1} \cdot 1 \right) = 2 \Rightarrow a_{22}^{(1)} = 2$



$$\begin{aligned}
 & a_{23}^{(1)} = \left( a_{23}^{(0)} - \frac{a_{21}^{(0)}}{a_{11}^{(0)}} a_{13}^{(0)} \right) = \left( 2 - \frac{1}{1} \cdot 1 \right) = 1 \Rightarrow a_{23}^{(1)} = 1 \\
 \bullet \quad i = 2 \text{ et } j = 3 & \Rightarrow b_2^{(1)} = \left( b_2^{(0)} - \frac{a_{21}^{(0)}}{a_{11}^{(0)}} b_1^{(0)} \right) = \left( 13 - \frac{1}{1} \cdot 6 \right) = 7 \Rightarrow b_2^{(1)} = 7 \\
 & a_{32}^{(1)} = \left( a_{32}^{(0)} - \frac{a_{31}^{(0)}}{a_{11}^{(0)}} a_{12}^{(0)} \right) = \left( 1 - \frac{1}{1} \cdot 1 \right) = 0 \Rightarrow a_{32}^{(1)} = 0 \\
 & a_{33}^{(1)} = \left( a_{33}^{(0)} - \frac{a_{31}^{(0)}}{a_{11}^{(0)}} a_{13}^{(0)} \right) = \left( -1 - \frac{1}{1} \cdot 1 \right) = -2 \Rightarrow a_{33}^{(1)} = -2 \\
 \bullet \quad i = 3 \text{ et } j = 3 & \Rightarrow b_3^{(1)} = \left( b_3^{(0)} - \frac{a_{31}^{(0)}}{a_{11}^{(0)}} b_1^{(0)} \right) = \left( 0 - \frac{1}{1} \cdot 6 \right) = -6 \Rightarrow b_3^{(1)} = -6
 \end{aligned}$$

Donc la forme du système à l'étape 1 est :

$$\begin{aligned}
 x_1 + x_2 + x_3 &= 6 && \text{equ(1)} \\
 0 \cdot x_1 + 2x_2 + x_3 &= 7 && \text{equ(2)} \\
 0 \cdot x_1 + 0 \cdot x_2 - 2x_3 &= -6 && \text{equ(3)}
 \end{aligned}$$

On remarque qu'au terme de l'étape 1, le système est déjà triangulaire. On n'a donc pas besoin de passer par l'étape 2. On saute directement à la résolution du système en calculant  $x_3$  à partir de l'equ(3), puis  $x_2$  de l'equ(2) et enfin  $x_1$  de l'equ(1).

**Exemple 2**

$$\begin{cases}
 0 \cdot x_1 + x_2 + 3x_3 = 1 \\
 x_1 + 2x_2 + 3x_3 = 4 \\
 6x_1 + 8x_2 + x_3 = 1
 \end{cases}$$

Dans ce cas, on remarque que le coefficient  $a_{11}^{(0)} = a_{11} = 0$ . Pour calcul les  $a_{ij}^{(1)}$ , il faut diviser par  $a_{11}$  qui est nul :  $a_{ij}^{(1)} = \left( a_{ij}^{(0)} - \frac{a_{i1}^{(0)}}{a_{11}^{(0)}} a_{1j}^{(0)} \right)$ .

Pour y remédier on a quelques astuces utiles à employer. Par exemple, On peut changer l'ordre des équations ce qui n'affecte pas la solution. Si on change l'ordre des équations (1) et (2) à (2) et (1), le système devient :

$$\begin{cases}
 x_1 + 2x_2 + 3x_3 = 4 \\
 0 \cdot x_1 + x_2 + 3x_3 = 1 \\
 6x_1 + 8x_2 + x_3 = 1
 \end{cases} \Rightarrow \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 3 \\ 6 & 8 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 4 \\ 1 \\ 1 \end{pmatrix}$$

On procède de la même manière que l'exemple précédent en commençant par l'étape 1 :

$$a_{ij}^{(1)} = \left( a_{ij}^{(0)} - \frac{a_{i1}^{(0)}}{a_{11}^{(0)}} a_{1j}^{(0)} \right) \quad \text{Et} \quad b_i^{(1)} = \left( b_i^{(0)} - \frac{a_{i1}^{(0)}}{a_{11}^{(0)}} b_1^{(0)} \right)$$

On commence le calcul de chaque élément avec  $i > p$  et  $j > p$  :

$$\begin{aligned}
 \bullet \quad i = 2 \text{ et } j = 2 & \Rightarrow a_{22}^{(1)} = \left( a_{22}^{(0)} - \frac{a_{21}^{(0)}}{a_{11}^{(0)}} a_{12}^{(0)} \right) = \left( 1 - \frac{0}{1} \times 2 \right) = 1 \Rightarrow a_{22}^{(1)} = 1 \\
 & a_{23}^{(1)} = \left( a_{23}^{(0)} - \frac{a_{21}^{(0)}}{a_{11}^{(0)}} a_{13}^{(0)} \right) = \left( 3 - \frac{0}{1} \times 3 \right) = 3 \Rightarrow a_{23}^{(1)} = 3 \\
 \bullet \quad i = 2 \text{ et } j = 3 & \Rightarrow b_2^{(1)} = \left( b_2^{(0)} - \frac{a_{21}^{(0)}}{a_{11}^{(0)}} b_1^{(0)} \right) = \left( 1 - \frac{0}{1} \times 4 \right) = 1 \Rightarrow b_2^{(1)} = 1 \\
 \bullet \quad i = 3 \text{ et } j = 2 & \Rightarrow a_{32}^{(1)} = \left( a_{32}^{(0)} - \frac{a_{31}^{(0)}}{a_{11}^{(0)}} a_{12}^{(0)} \right) = \left( 8 - \frac{6}{1} \times 2 \right) = -4 \Rightarrow a_{32}^{(1)} = -4
 \end{aligned}$$

$$\begin{aligned}
 a_{33}^{(1)} &= \left( a_{33}^{(0)} - \frac{a_{31}^{(0)}}{a_{11}^{(0)}} a_{13}^{(0)} \right) = \left( 1 - \frac{6}{1} \times 3 \right) = -17 \Rightarrow a_{33}^{(1)} = -17 \\
 \bullet \quad i = 3 \text{ et } j = 3 &\Rightarrow b_3^{(1)} = \left( b_3^{(0)} - \frac{a_{31}^{(0)}}{a_{11}^{(0)}} b_1^{(0)} \right) = \left( 1 - \frac{6}{1} \times 4 \right) = -23 \Rightarrow b_3^{(1)} = -23
 \end{aligned}$$

Donc la forme du système à l'étape 1 est :

$$\begin{aligned}
 x_1 + 2x_2 + 3x_3 &= 4 && \text{equ(1)} \\
 0x_1 + x_2 + 3x_3 &= 1 && \text{equ(2)} \\
 0x_1 - 4x_2 - 17x_3 &= -23 && \text{equ(3)}
 \end{aligned}$$

Il n'est pas encore triangulaire. On passe alors à l'étape 2, le calcul des coefficients  $a_{ij}^{(2)}$  et  $b_i^{(2)}$

$$a_{ij}^{(2)} = \left( a_{ij}^{(1)} - \frac{a_{i2}^{(1)}}{a_{22}^{(1)}} a_{2j}^{(1)} \right) \quad \text{Et} \quad b_i^{(2)} = \left( b_i^{(1)} - \frac{a_{i2}^{(1)}}{a_{22}^{(1)}} b_2^{(1)} \right)$$

Il faut se rappeler que les coefficients  $a_{ij}^{(p)}$  tel que  $i \leq p$  ou  $j \leq p$  sont nuls donc ce n'est pas la peine de les calculer. Pour  $p = 2$ , il reste seulement  $a_{33}^{(2)}$  et  $b_3^{(2)}$

$$\begin{aligned}
 a_{33}^{(2)} &= \left( a_{33}^{(1)} - \frac{a_{32}^{(1)}}{a_{22}^{(1)}} a_{23}^{(1)} \right) = \left( -17 - \frac{-4}{1} \cdot 3 \right) = -5 \Rightarrow a_{33}^{(2)} = -5 \\
 i = 3 \text{ et } j = 3 &\Rightarrow b_3^{(2)} = \left( b_3^{(1)} - \frac{a_{32}^{(1)}}{a_{22}^{(1)}} b_2^{(1)} \right) = \left( -23 - \frac{-4}{1} \cdot 1 \right) = -19 \Rightarrow b_3^{(2)} = -19
 \end{aligned}$$

Ainsi le système à l'étape 2 devient :

$$\begin{aligned}
 x_1 + 2x_2 + 3x_3 &= 4 && \text{equ(1)} \\
 0x_1 + x_2 + 3x_3 &= 1 && \text{equ(2)} \\
 0x_1 + 0x_2 - 5x_3 &= -19 && \text{equ(3)}
 \end{aligned}$$

La solution du système est :  $\left\{ x_3 = \frac{19}{5}; x_2 = -\frac{52}{5}; x_1 = \frac{67}{5} \right\}$

**Exemple 3**

$$\begin{aligned}
 x_1 + x_2 + x_3 + x_4 &= 1 \\
 2x_1 + x_2 + x_3 - x_4 &= 1 \\
 x_1 + 2x_2 + x_3 + 2x_4 &= 1 \\
 x_1 - x_2 - x_3 + x_4 &= 1
 \end{aligned}
 \Rightarrow \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & -1 \\ 1 & 2 & 1 & 2 \\ 1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

C'est un système  $4 \times 4$ . On a besoin de 3 étapes en générale pour triangulariser le système en procédant de la même manière que les exemples précédents.

En générale, la forme finale (triangulaire) du système qui nous permettra de le résoudre est :

$$\begin{aligned}
 x_1 + x_2 + x_3 + x_4 &= 1 \\
 0x_1 + a_{22}^{(1)} x_2 + a_{23}^{(1)} x_3 + a_{24}^{(1)} x_4 &= b_2^{(1)} \\
 0x_1 + 0x_2 + a_{33}^{(2)} x_3 + a_{34}^{(2)} x_4 &= b_3^{(2)} \\
 0x_1 + 0x_2 + 0x_3 + a_{44}^{(3)} x_4 &= b_4^{(3)}
 \end{aligned}$$

- **A l'étape 1**, le système s'écrit avec les coefficients :

$$a_{ij}^{(1)} = \left( a_{ij}^{(0)} - \frac{a_{i1}^{(0)}}{a_{11}^{(0)}} a_{1j}^{(0)} \right) \quad \text{et} \quad b_i^{(1)} = \left( b_i^{(0)} - \frac{a_{i1}^{(0)}}{a_{11}^{(0)}} b_1^{(0)} \right) \quad i, j = 2, \dots, 4$$

$$\begin{aligned} x_1 + x_2 + x_3 + x_4 &= 1 \\ 0x_1 - x_2 - x_3 - 3x_4 &= -1 \\ 0x_1 + x_2 + 0x_3 + x_4 &= 0 \\ 0x_1 - 2x_2 - 2x_3 + 0x_4 &= 0 \end{aligned}$$

- A l'étape 2 :

$$a_{ij}^{(2)} = \left( a_{ij}^{(1)} - \frac{a_{i2}^{(1)}}{a_{22}^{(1)}} a_{2j}^{(1)} \right) \quad \text{et} \quad b_i^{(2)} = \left( b_i^{(1)} - \frac{a_{i2}^{(1)}}{a_{22}^{(1)}} b_2^{(1)} \right) \quad i, j = 3, 4$$

$$\begin{aligned} x_1 + x_2 + x_3 + x_4 &= 1 \\ 0x_1 - x_2 - x_3 - 3x_4 &= -1 \\ 0x_1 + 0x_2 - x_3 - 2x_4 &= -1 \\ 0x_1 + 0x_2 + 0x_3 + 6x_4 &= 2 \end{aligned}$$

Ici encore on n'a pas besoin de passer par l'étape 3 car au terme de l'étape 2 le système est déjà triangulaire. La solution du système est :  $\left\{ x_4 = \frac{1}{3}; x_3 = \frac{1}{3}; x_2 = -\frac{1}{3}; x_1 = \frac{2}{3} \right\}$ .

## 2.2.4 Programme de la méthode « Elimination de Gauss »

### 2.2.4.1 Système à $3 \times 3$

```

implicit none
integer :: i,j,p
doubleprecision,dimension(3,3,0:2):: a
doubleprecision,dimension(3,0:2):: b
doubleprecision,dimension(3):: x
do i=1,3 !Entrée des coefficients aij
do j=1,3
write(*,*) ' donner l'element a(',i,j,',0)'
read(*,*) a(i,j,0)
enddo
enddo
do i=1,3 !Entrée des bij
write(*,*) ' donner l'element b(',i,', 0)'
read(*,*) b(i,0)
enddo
do p=1,2 !Calcul des coefficients de chaque étape p
do j=1,3
do i=1,3
if(i.gt.p.and.j.gt.p)then
a(i,j,p)=a(i,j,p-1)-(a(i,p,p-1)/a(p,p,p-1))*a(p,j,p-1)
endif
enddo
enddo
enddo
do i=1,3
do p=1,2
b(i,p)=b(i,p-1)-(a(i,p,p-1)/a(p,p,p-1))*b(p,p-1)
enddo
enddo
!Calcul et affichage de la solution X={x(1); x(2); x(3)}
x(3)=b(3,2)/a(3,3,2)
x(2)=(b(2,1)-a(2,3,1)*x(3))/a(2,2,1)
x(1)=(b(1,0)-a(1,2,0)*x(2)-a(1,3,0)*x(3))/a(1,1,0)
write(*,*) 'la solution x1,x2,x3 est:'
write(*,*) 'x1=',x(1),' x2=',x(2),' x3=',x(3)
end

```

Figure 2.1 : Programme de la méthode Elimination de Gauss pour un système  $3 \times 3$

2.2.4.2 Système à  $N \times N$ 

```

implicit none
integer :: i,j,p, n=10 !le nombre d'équ. n peut être modifié ici
doubleprecision,dimension(10,10,0:10-1):: a
doubleprecision,dimension(10,0:2):: b
doubleprecision,dimension(10):: x
doubleprecision::s
do i=1,n !Entrée des coefficients aij
do j=1,n
write(*,*) ' donner l'element a(',i,j,',0)'
read(*,*) a(i,j,0)
enddo
enddo
do i=1,n !Entrée des bij
write(*,*) ' donner l'element b(',i,', 0)'
read(*,*) b(i,0)
enddo
do p=1,n-1 !Calcul des coefficinetns de chaque étape p
do j=1,n
do i=1,n
if(i.gt.p.and.j.gt.p)then
a(i,j,p)=a(i,j,p-1)-(a(i,p,p-1)/a(p,p,p-1))*a(p,j,p-1)
endif
enddo
enddo
enddo
do i=1,n
do p=1,n-1
b(i,p)=b(i,p-1)-(a(i,p,p-1)/a(p,p,p-1))*b(p,p-1)
enddo
enddo
s=0 !Calcul et affichage de la solution X={x(i): i=1,...,n}
do p=n-1,1
do i=n,1
s=s+a(i,i+1,p)*x(i+1)
x(i)=(b(i,p)-s)/a(i,i,p)
write(*,*) "x", i, "=", x(i)
enddo
enddo

```

Figure 2.2 : Programme de la méthode Elimination de Gauss pour un système  $N \times N$ 

Exercice

Soient les systèmes d'équations linéaires suivantes :

$$\begin{array}{lll}
 x_1 + x_2 + x_3 = 6 & x_1 + x_2 + x_3 + x_4 = 1 & 0 \cdot x_1 + x_2 + 3x_3 = 1 \\
 x_1 + 3x_2 + 2x_3 = 13 & 2x_1 + x_2 + x_3 - x_4 = 1 & x_1 + 2x_2 + 3x_3 = 4 \\
 x_1 + x_2 - x_3 = 0 & x_1 + 2x_2 + x_3 + 2x_4 = 1 & 6x_1 + 8x_2 + x_3 = 1 \\
 & x_1 - x_2 - x_3 + x_4 = 1 &
 \end{array}$$

Réécrire ces système linéaires sous forme matricielle  $AX = B$  puis trouver leurs solutions en utilisant la méthode d'élimination de Gauss en écrivant la nouvelle forme du système à chaque étape.

## 2.3 Décomposition LU (factorisation)

### 2.3.1 Définition

Pour un système d'équations linéaires de la forme  $AX=B$ , il existe une matrice triangulaire inférieure  $L$  avec  $L_{ii} = 1$  ( $i = 1, \dots, n$ ) et une matrice triangulaire supérieure  $U$  telles que  $A=LU$ . En particulier, toute matrice inversible admet une factorisation LU.

### 2.3.2 Théorème

Soit  $A$  une matrice inversible d'ordre  $n$  dont **les mineurs principaux** sont non nuls (condition nécessaire et suffisante). Alors il existe une unique matrice  $L$  triangulaire inférieure avec des 1 sur la diagonale, et une unique matrice  $U$  triangulaire supérieure telles que  $A = LU$ .

Rappelons que le mineur principal d'ordre  $i$  de  $A$  est le déterminant des  $i$  premières lignes et  $i$

$$\text{premières colonnes tel que : } (det(A))_i = \begin{vmatrix} a_{11} & \cdots & a_{1i} \\ \vdots & \ddots & \vdots \\ a_{i1} & \cdots & a_{ii} \end{vmatrix}$$

### 2.3.3 L'algorithme

On calcule les matrices  $L$  et  $U$  telles que  $A=LU$  par les formules déduites de la manière suivante :

$$\begin{aligned} U_{1j} &= a_{1j} \text{ si } j = 1 \dots n & L_{ii} &= 1 & L_{i1} &= \frac{a_{ij}}{U_{jj}} \text{ si } i = 1 \dots n \\ U_{ij} &= 0 \text{ si } j = 1 \dots i - 1 & & & L_{ij} &= 0 \text{ si } j = i + 1 \dots n \\ U_{ij} &= a_{ij} - \sum_{k=1}^{i-1} L_{ik}U_{kj} & L_{ij} &= \left( a_{ij} - \sum_{k=1}^{i-1} L_{ik}U_{kj} \right) / U_{jj} \end{aligned}$$

En remplaçant  $A$  par  $LU$ , le système  $AX = B$  devient  $L(UX)=B$  on remarque que si on pose  $Y=UX$  le système s'écrit alors  $LY = B$ , le système est alors facilement résolu pour déterminer  $Y$ . Puis on résout le system  $UX=Y$  ce qui donne les valeurs de  $X$ .

#### Exemple

Résoudre, en utilisant la méthode de décomposition LU, le système d'équations linéaires suivant :

$$\begin{cases} 9x_1 + 6x_2 + 3x_3 = 1 \\ 6x_1 + 3x_2 + x_3 = 2 \\ x_1 + x_3 = 3 \end{cases} \equiv AX = B \equiv \begin{pmatrix} 9 & 6 & 3 \\ 6 & 3 & 1 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

La décomposition LU consiste à résoudre le système après avoir transformé la matrice  $A$  sous la forme d'un produit de deux matrices triangulaires tel que :  $A = L \times U$

Avec  $L$  : une matrice inférieure (Lower) et  $U$  : une matrice supérieure (Upper)

Le nouveau système devient :  $\begin{cases} LY = B \\ UX = Y \end{cases}$  Il suffit alors de résoudre le système  $LY = B$  en premier (trouver les  $y_i$ ) pour enfin déterminer les  $x_i$  à partir du système  $UX = Y$

#### i. Détermination des $U_{ij}$ et $L_{ij}$

On applique les relations posées précédemment dans l'algorithme. Nous avons un système  $3 \times 3$  c.à.d.  $i$  et  $j$  varient de 1 à 3.

Pour  $\forall j = 1, \dots, 3$  et quand  $i = 1 \Rightarrow U_{1j} = a_{1j}$  ce qui donne :  $U_{11} = 9$  ;  $U_{12} = 6$  ;  $U_{13} = 3$

Pour  $\forall i = 1, \dots, 3$

Pour  $j < i \Rightarrow U_{ij} = 0$  ce qui implique :  $U_{21} = U_{31} = U_{32} = 0$

Pour  $j > i \Rightarrow L_{ij} = 0$  ce qui donne :  $L_{12} = L_{13} = L_{23} = 0$

$$\text{Quand } j = 1 \Rightarrow L_{i1} = \frac{a_{ij}}{U_{jj}} \Rightarrow \begin{cases} L_{11} = \left( \frac{a_{11}}{U_{11}} \right) = \frac{9}{9} = 1 \\ L_{21} = \left( \frac{a_{21}}{U_{11}} \right) = \frac{6}{9} = \frac{2}{3} = 0,667 \\ L_{31} = \left( \frac{a_{31}}{U_{11}} \right) = \frac{1}{9} = 0,111 \end{cases}$$

Pour  $i = j \Rightarrow L_{ii} = 1$  ceci implique que :  $L_{11} = L_{22} = L_{33} = 1$

Il manque un élément pour la matrice L ( $L_{32}$ ) et trois éléments de la matrice U ( $U_{22}$  ;  $U_{23}$  et  $U_{33}$ ) on les détermine à partir des formules de  $L_{ij}$  et  $U_{ij}$  citées plus haut.

$$U_{22} = A_{22} - \sum_{k=1}^{2-1=1} L_{2k}U_{k2} = A_{22} - L_{21}U_{12} = 3 - \left(\frac{2}{3}\right) \times 6 \Rightarrow U_{22} = -1$$

$$L_{32} = \left( A_{32} - \sum_{k=1}^{3-1=2} L_{3k}U_{k2} \right) / U_{22} = (A_{32} - (L_{31}U_{12} + L_{32}U_{22})) / U_{22}$$

$$= (A_{32} - L_{31}U_{12}) / U_{22} - L_{32}$$

$$\Rightarrow 2L_{32} = (A_{32} - L_{31}U_{12}) / U_{22} = \left( 0 - \left( \frac{1}{9} \right) \times 6 \right) / (-1) = \frac{2}{3} \Rightarrow L_{32} = \frac{1}{3} = 0,667$$

$$U_{23} = A_{23} - L_{21}U_{13} = 1 - \frac{2}{3} \times 3 \Rightarrow U_{23} = -1$$

$$U_{33} = A_{33} - \sum_{k=1}^{3-1=2} L_{3k}U_{k3} = A_{33} - L_{31}U_{13} - L_{32}U_{23} = 1 - 0,111 \times 3 - 0,667 \times (-1)$$

$$\Rightarrow U_{33} = 1,333$$

On regroupe les résultats et on obtient les matrices triangulaires requises pour la résolution du système :

$$U = \begin{pmatrix} 9 & 6 & 3 \\ 0 & -1 & -1 \\ 0 & 0 & 1,333 \end{pmatrix} \quad \text{et} \quad L = \begin{pmatrix} 1 & 0 & 0 \\ 0,667 & 1 & 0 \\ 0,111 & 0,667 & 1 \end{pmatrix}$$

**ii. Résolution du système LY=B**

$$LY = B \Rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0,667 & 1 & 0 \\ 0,111 & 0,667 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

Pour résoudre ce système il suffit de déterminer les  $y_i$  par descente comme suit :

$$\begin{cases} y_1 = \frac{b_1}{L_{11}} = -\frac{1}{1} & \Rightarrow y_1 = 1 \\ y_2 = \frac{b_2 - L_{21}y_1}{L_{22}} = \frac{2 - 0,667 \times 1}{1} = \frac{4}{3} & \Rightarrow y_2 = 1,333 \\ y_3 = \frac{b_3 - L_{31}y_1 - L_{32}y_2}{L_{33}} = \frac{3 - 0,111 \times 1 - 0,667 \times 1,333}{1} & \Rightarrow y_3 = 2 \end{cases}$$

**iii. Détermination de la solution X du système linéaire AX=B à partir de UX=Y**

$$UX = Y \Rightarrow \begin{pmatrix} 9 & 6 & 3 \\ 0 & -1 & -1 \\ 0 & 0 & 1,333 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1,333 \\ 2 \end{pmatrix}$$

Dans ce cas, on détermine les  $x_i$  par remontée comme suit :

$$\begin{cases} x_3 = \frac{y_3}{U_{33}} = \frac{2}{1,333} & \Rightarrow x_3 = 1,50 \\ x_2 = \frac{y_2 - U_{23}x_3}{U_{22}} = \frac{1,333 - (-1) \times 1,50}{-1} & \Rightarrow x_2 = -2,83 \\ x_1 = \frac{y_1 - U_{12}x_2 - U_{13}x_3}{U_{11}} = \frac{1 - 6 \times (-2,83) - 3 \times 1,50}{9} & \Rightarrow x_1 = 1,50 \end{cases}$$

Enfin la solution du système linéaire est  $x = \begin{pmatrix} 1,50 \\ -2,83 \\ 1,50 \end{pmatrix}$

Pour faciliter ces étapes et avoir le résultat en moins de temps, on utilise le programme FORTRAN.

## 2.3.4 Programme de la méthode de « Décomposition LU »

```

Program LU_decomp
Implicit None
integer:: i, j, k, p, n=3
double precision, dimension(1:3,1:3):: a, l, u, s1, s2
double precision, dimension(1:3):: x, b, y
double precision :: sx=0, sy=0

do i=1,n
do j=1,n
write(*,*) "donnez l'element a(", i, j, ") de la matrice A"
read(*,*) a(i,j)
enddo
enddo

do j=1,n
write(*,*) "donnez l'element b(", j, ") de la matrice B"
read(*,*) b(j)
enddo

!Calcul des u(i,j) et l(i,j)
do i=1,n
do j=1,n
if(i==j)then
l(i,j)=1
endif
if(i==1)then
u(i,j)=a(i,j)
endif
if(j<i)then
u(i,j)=0
endif
if(j>i)then
l(i,j)=0
endif
if(j==1)then
l(i,1)=(a(i,j)/u(j,j))
endif
s1(i,j)=0
s2(i,j)=0
enddo
enddo

do i=1,n
do j=1,n
if(i>1.and.j>(i-1))then
do k=1,i-1
s1(i,j)=s1(i,j)+(l(i,k)*u(k,j))
enddo
u(i,j)=a(i,j)-s1(i,j)
endif
if(j>1.and.j<(i+1).and.i.ne.j)then
do p=1,j-1
s2(i,j)=s2(i,j)+(l(i,p)*u(p,j))
enddo
l(i,j)=(a(i,j)-s2(i,j))/u(j,j)
endif
enddo

!calcul des y(i) par descente
y(1)=b(1)/l(1,1)
write(*,*) "y(", 1, ")=", y(1)
do i=2,n
sy=0.0
do j=1,i-1
sy=sy+l(i,j)*y(j)
enddo
y(i)=(b(i)-sy)/l(i,i)
write(*,*) "y(", i, ")=", y(i)
enddo

!calcul des x(i) par remontée
x(n)=y(n)/u(n,n)
write(*,*) "x(", n, ")=", x(n)
do i=1,n-1
sx=0.0
do j=1,n-1
sx=sx+u(n-i,n-j+1)*x(n-j+1)
enddo
x(n-i)=(y(n-i)-sx)/u(n-i,n-i)
write(*,*) "x(", n-i, ")=", x(n-i)
enddo
end

```

```

F:\Software Setups\Fortran90\F90\LU_decomp2.exe
donnez l'element a( 1 1)
9
donnez l'element a( 1 2)
6
donnez l'element a( 1 3)
3
donnez l'element a( 2 1)
6
donnez l'element a( 2 2)
3
donnez l'element a( 2 3)
1
donnez l'element a( 3 1)
1
donnez l'element a( 3 2)
0
donnez l'element a( 3 3)
1
donnez l'element b( 1)
1
donnez l'element b( 2)
2
donnez l'element b( 3)
3
y( 1)= 1.000000000000000
y( 2)= 1.3333333333333334
y( 3)= 2.000000000000000
x( 3)= 1.500000000000000
x( 2)= -2.8333333333333334
x( 1)= 1.500000000000000
Press any key to continue

```

Figure 2.3 : Programme de la méthode de Décomposition LU avec la solution de l'exemple



### 2.4 Méthode Itérative de Jacobi

#### 2.4.1 Définition

L'idée des méthodes itératives est de construire une suite de vecteurs  $X^{(k)}$  qui converge vers le vecteur  $X$ , solution du système  $AX = B$ . On s'arrête au bout d'un nombre fini d'itérations  $n$  choisi pour que  $X^{(n)}$  soit suffisamment voisin de  $X$ .

Un système initial d'équations linéaires est transformé de tel sorte que chaque composante  $x_i$  est écrite en fonction des autres et du  $b_i$  comme démontré dans ce qui suit :

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n = b_2 \\ \dots \dots \dots \\ a_{p1}x_1 + a_{p2}x_2 + a_{p3}x_3 + \dots + a_{pn}x_n = b_p \\ \dots \dots \dots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n = b_n \end{cases} \Rightarrow \begin{cases} x_1 = \frac{b_1 - a_{12}x_2 - a_{13}x_3 - \dots - a_{1n}x_n}{a_{11}} \\ x_2 = \frac{b_2 - a_{21}x_1 - a_{22}x_2 - \dots - a_{2n}x_n}{a_{22}} \\ \dots \dots \dots \\ x_i = \frac{b_i - a_{i1}x_1 - a_{i2}x_2 - \dots - a_{in}x_n}{a_{ii}} \\ \dots \dots \dots \\ x_n = \frac{b_n - a_{n1}x_1 - a_{n2}x_2 - \dots - a_{n(n-1)}x_{(n-1)}}{a_{nn}} \end{cases}$$

En généralisant, on obtient :

$$\begin{cases} x_1 = \frac{b_1 - \sum_{j \neq 1} a_{1j}x_j}{a_{11}} \\ x_2 = \frac{b_2 - \sum_{j \neq 1} a_{2j}x_j}{a_{22}} \\ \dots \dots \dots \\ x_i = \frac{b_i - \sum_{j \neq i} a_{ij}x_j}{a_{ii}} \\ \dots \dots \dots \\ x_n = \frac{b_n - \sum_{j \neq n} a_{nj}x_j}{a_{nn}} \end{cases} \Rightarrow x_i = \left( b_i - \sum_{j=1; i \neq j}^n a_{ij}x_j \right) / a_{ii} \quad (\text{II. 9})$$

#### 2.4.2 L'algorithme

Cette méthode est basée sur le même principe de la méthode du point fixe  $x=g(x)$ . A partir de la valeur initiale  $X^0$  on calcul la nouvelle valeur de  $X$  et on répète le même procédé itératif.

$$x_i^k = \left( b_i - \sum_{\substack{j=1 \\ i \neq j}}^n a_{ij}x_j^{k-1} \right) / a_{ii} \quad \equiv \quad X = g(X) \quad (\text{II. 10})$$

En notant  $X^{(k)}$  le vecteur de composantes  $X^{(k)} = (x_1^k, x_2^k, x_3^k, \dots, x_n^k)$  :

- L'algorithme est initialisé par un vecteur arbitraire  $x^{(0)} = (x_1^0, x_2^0, x_3^0, \dots, x_n^0)$
- Il effectue ses itérations suivant la formule  $x_i = (b_i + \sum_{j=1; i \neq j}^n a_{ij}x_j) / a_{ii}$
- Il s'arrête quand  $|x_i^k - x_i^{k-1}| < \epsilon ; \forall i = 1..n$  où  $\epsilon$  est une valeur de précision donnée

Lorsque la suite  $X^{(k)}$  converge, i.e.  $\lim_{k \rightarrow +\infty} X^{(k)} = X^{(S)}$ , fini, on dit que la méthode converge.

2.4.3 Théorème

La méthode ne converge pas toujours. Si A est une matrice à diagonale strictement dominante :

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| \quad (1 \leq i \leq n) \tag{II. 11}$$

alors la méthode itérative de Jacobi appliquée au système AX=B est convergente pour tout  $X^{(0)}$ , Exemple

On considère le système d'équations linéaires suivant :

$$\begin{cases} 4x_1 - x_2 = 6 \\ -x_1 + 4x_2 - x_3 = 4 \\ -x_2 + 4x_3 = 6 \end{cases}$$

- a) Montrer que la méthode de Jacobi est convergente dans ce cas
- b) Sachant que la solution exacte de ce système est  $X = (2,2,2)$  et en utilisant la méthode itérative de Jacobi avec  $X^{(0)} = (0,0,0)$ , calculer les cinq premières itérations et déterminer la solution X du système et comparer les résultats avec la solution exacte.

**Solution**

**a) Convergence**

On remarque que A est une matrice à diagonale strictement dominante car ces éléments diagonaux sont supérieurs à la somme des éléments non diagonaux :  $|a_{ii}| \geq \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| \Rightarrow |(-1 - 1 + 0)| = 2 < |a_{ii}|$

Ceci implique que **la méthode de Jacobi est convergente.**

**b) Application de la méthode itérative de Jacobi**

Effectuons cinq itérations avec cette méthode c.à.d.  $k = 1, 2, \dots, 5$ . On calcule  $X^{(1)}$  à partir de  $X^{(0)}$  en procédant de la manière suivante :  $x_i^{(1)} = \left( b_i - \sum_{\substack{j=1 \\ i \neq j}}^3 a_{ij} x_j^{(0)} \right) / a_{ii}$

$$x_1^{(1)} = \left( b_1 - \sum_{\substack{j=1 \\ i \neq j}}^3 a_{1j} x_j^{(0)} \right) / a_{11} = \left( b_1 - (a_{12} \times x_2^{(0)} + a_{13} \times x_3^{(0)}) \right) / a_{11} \Rightarrow x_1^{(1)} = 1,5$$

$$x_2^{(1)} = \left( b_2 - \sum_{\substack{j=1 \\ i \neq j}}^3 a_{2j} x_j^{(0)} \right) / a_{22} = \left( b_2 - (a_{21} \times x_1^{(0)} + a_{23} \times x_3^{(0)}) \right) / a_{22} \Rightarrow x_2^{(1)} = 1$$

$$x_3^{(1)} = \left( b_3 - \sum_{\substack{j=1 \\ i \neq j}}^3 a_{3j} x_j^{(0)} \right) / a_{33} = \left( b_3 - (a_{31} \times x_1^{(0)} + a_{32} \times x_2^{(0)}) \right) / a_{33} \Rightarrow x_3^{(1)} = 1,5$$

On continue de cette manière pour calculer  $X^{(2)}$  à partir de  $X^{(1)}$ , puis  $X^{(3)}$  à partir de  $X^{(2)}$ , ainsi de suite jusqu'à atteindre la cinquième itération  $X^{(5)}$  (voir le tableau ci-dessous).

Tableau 2.1 : Résultats de calcul en appliquant la Méthode Itérative de Jacobi

$a_{i,1}$	$a_{i,2}$	$a_{i,3}$	$b_j$	$x^0$	$x_i^{(1)}$	$x_i^{(2)}$	$x_i^{(3)}$	$x_i^{(4)}$	$x_i^{(5)}$
4	-1	0	6	0	1,5	1,75	1,9375	1,96875	<b>1,9921875</b>
-1	4	-1	4	0	1	1,75	1,875	1,96875	<b>1,984375</b>
0	-1	4	6	0	1,5	1,75	1,9375	1,96875	<b>1,9921875</b>

On remarque que la solution obtenue par la méthode de Jacobi converge bien vers la solution exacte. Si on augmente le nombre d'itérations, la solution numérique s'approchera d'autant plus à la solution exacte. Pour cela on utilise le programme FORTRAN pour imposer un nombre assez grand d'itérations et avoir le résultat en un seul click.

#### 2.4.4 Programme de la « Méthode Itérative de Jacobi »

```

Program Iterative_Method
implicit none
real, dimension (1000):: b, x, s, x0
real, dimension (1000,1000):: a
integer:: i, j, n, k, p, l, m

!Entrée des données
write(*,*) "donnez la dimension du système n"
read(*,*) n

write(*,*) "entrez les coefficients a(i,j)"
do k=1, n
  do p=1, n
    write(*,*) "entrez l'element", k,p
    read(*,*) a(k,p)
  enddo
enddo

write(*,*) "entrez les données b(i)"
do l=1, n
  write(*,*) "entrez l'element b", l
  read(*,*) b(l)
enddo

write(*,*) "entrez les données initiales X(0)"
do k=1, n
  read(*,*) x0(k)
enddo

m=0
4 do i=1, n
  s(i)=0
  do j=1,n
    if(i/=j) then
      s(i)=s(i)+a(i,j)*x0(j)
    else
      s(i)=s(i)
    endif
  enddo
  !print*,i,s(i)
  x(i)=(b(i)-s(i))/a(i,i)
enddo

do i=1, n
  x0(i)=x(i)
enddo
  m=m+1
  if(m<5) goto 4
do i=1, n
write(*,*) "x(", i, ")=", x(i)
enddo
End

```

```

F:\Software Setups\Fortran90\F90\Iterative_Method1.exe
donnez la dimension du systpme n
3
entrez les coefficients a(i,j)
entrez l'element      1      1
4
entrez l'element      1      2
-1
entrez l'element      1      3
0
entrez l'element      2      1
-1
entrez l'element      2      2
4
entrez l'element      2      3
-1
entrez l'element      3      1
0
entrez l'element      3      2
-1
entrez l'element      3      3
4
entrez les données b(i)
entrez l'element b      1
6
entrez l'element b      2
4
entrez l'element b      3
6
entrez les données initiales X(0)
0
0
0
x(      1)=      1.992188
x(      2)=      1.984375
x(      3)=      1.992188
Press any key to continue

```

Figure 2.4 : Programme de la méthode Itérative de Jacobi avec solution de l'exemple à 5 itérations

**Remarque :** Si on augmente le nombre d'itérations à 100, par exemple, la solution numérique coïncide avec la solution exacte comme il est illustré sur la figure suivante.

```
x( 1)= 2.000000  
x( 2)= 2.000000  
x( 3)= 2.000000  
Press any key to continue
```

*Figure 2.5 : Méthode de Jacobi - Solution de l'exemple après 100 itérations*

**Chapitre 3**  
**INTEGRATIONS NUMERIQUES**

---

## Chapitre 3 INTEGRATIONS NUMERIQUES

### 3.1 Introduction

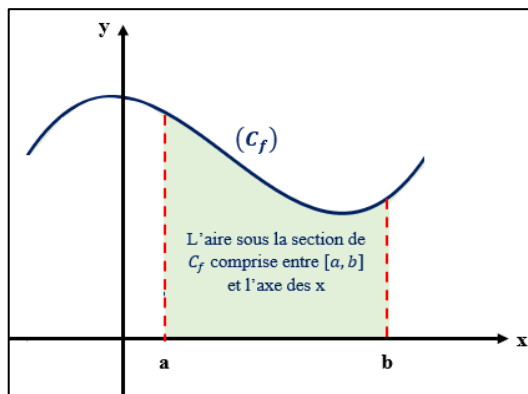
Considérée comme un passage du discret au continu, l'intégrale en physique est utilisée pour sommer les contributions d'éléments que l'on ne peut pas compter car leur distribution est continue (le long d'une ligne, sur un plan ou une nappe, ou même dans un volume) et non discrète (ensemble de points indénombrable).

### 3.2 Définition « Intégrale »

Soit une fonction  $f$  continue sur un intervalle  $[a, b]$ . L'intégrale de  $f$  sur cet intervalle  $[a, b]$  est définie comme étant l'aire comprise entre la courbe de  $(C_f)$ , les droites  $x = a$  et  $x = b$  et l'axe des abscisses  $x$ .

$$I = \int_a^b f(x)dx = \text{Surface} \quad (\text{III. 1})$$

Comme illustré sur le schéma de la figure 3.1



*Figure 3.1 : Schéma de la surface entre  $C_f$ , l'axe des  $x$  et l'intervalle  $[a, b]$*

Les méthodes numériques d'intégration interviennent alors lorsque le calcul analytique devient trop complexe. On les appelle les « Intégrales Numériques ».

### 3.3 Méthode des Rectangles

#### 3.3.1 Définition

La méthode des rectangles consiste à diviser l'intervalle  $[a, b]$  en sous-intervalles de largeurs égales ( $h = \Delta x = \frac{b-a}{n}$ ). Le calcul de l'intégrale est approché par la sommation de rectangles fins de largeur  $\Delta x$  et de hauteur  $f(x)$ . Le résultat de cette approximation est d'autant meilleur que la largeur  $\Delta x$  des rectangles tend vers zéro, c.à.d.  $\lim_{h \rightarrow 0} I_{rect} = I$ . Le schéma de la figure 3.2 illustre bien cette méthode.

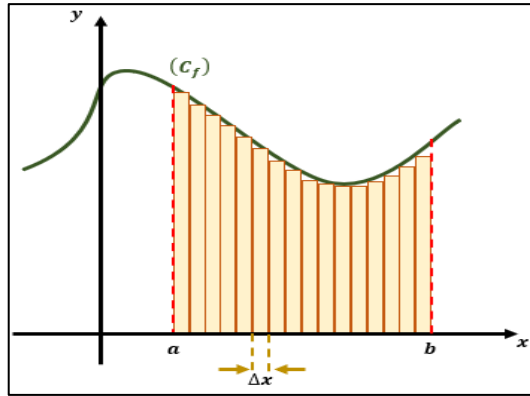


Figure 3.2 :Schéma représentant la méthode des rectangles

### 3.3.2 Théorème

On considère une fonction  $f$  continue sur l'intervalle  $[a, b]$  et dérivable sur l'intervalle ouvert  $]a, b[$ . On divise cet intervalle en  $x_i$  points équidistants (avec  $i = 0, \dots, n$ ). On note  $h$  le pas de cette subdivision.

La méthode des rectangles est une méthode d'ordre 0 car la fonction  $f$  est remplacée par une fonction constante par morceaux (constante dans l'intervalle  $[x_i ; x_{i+1}]$ ).

$$\int_a^b f(x)dx = \sum_{i=0}^{n-1} f(x_i)\Delta x = \sum_{i=0}^{n-1} f(x_i) \frac{(b-a)}{n} \quad (\text{III. 2})$$

Avec  $n$  le nombre de points de discrétisation et  $h = \Delta x = \frac{b-a}{n}$   
 $x_i = a + i \times h$

L'intégrale de la fonction  $f$  sur l'intervalle  $[a, b]$  se calcul alors comme suit :

$$I = h \times \sum_{i=0}^{n-1} f(a + i \times h) \quad (\text{III. 3})$$

Lorsque la subdivision se réduit à sa plus simple expression  $x_0 = a ; x_1 = b$ , on a :

$$\int_a^b f(x)dx \simeq (b-a)f(a) \quad (\text{III. 4})$$

Lorsque la première dérivée de  $f$  est bornée par une constante  $M$ , l'erreur  $\varepsilon$  dans la méthode des rectangles est estimée par la formule qui suit :

$$\varepsilon = |I - I_{rect}| \leq \frac{1}{2} \frac{(b-a)^2}{n} \sup |f'(x)|_{x \in [a,b]} \quad (\text{III. 5})$$

$I$  : intégrale de  $f$  calculée analytiquement et  $I_{rect}$  : intégrale de  $f$  calculée avec la méthode des rectangles.

#### Remarque Importante\*\*\*

On remarque que l'erreur estimée est de l'ordre de  $1/n$ . Pour assurer une bonne convergence du calcul, il faut prendre le plus grand nombre de sous-intervalles  $n$ .

3.3.3 L’algorithme

Les étapes de résolution par la méthode des rectangles sont très simples, telles que :

- Le choix du nombre de discrétisation  $n$  et ainsi le pas  $h$
- L’initialisation de la somme, qui représente le résultat de l’intégrale, à zéro  $s = 0$
- Lancement d’une boucle allant de 0 à  $n - 1$  pour faire :
  - Le calcul des points de discrétisation  $x_i$  dans l’intervalle  $[a, b]$
  - Le produit de  $f(x_i) \times h$
  - Ajouter le résultat du produit à la somme  $s$
  - Tester si la précision est atteinte (en utilisant la formule de  $\varepsilon$ ) : si oui => sortir de la boucle, si non => continuer la boucle
- Affichage du dernier résultat de la somme  $s$  (résultat de l’intégrale).

Exemple

Soit la fonction  $f$  définie sur  $\mathbb{R}$  par :  $f(x) = \cos x$ . **N.B.** On cherche à déterminer l’intégrale analytique (la valeur) de la fonction, pas l’intégrale graphique (sa surface).

1. Calculer, à l’aide de la méthode des rectangles, une valeur approchée de  $I$  ( $I = \int_{-\pi}^{\pi/2} f(x) dx$ ) en utilisant  $n = 10$  sous-intervalles.
2. Sachant que la valeur exacte de l’intégrale est ( $I = \int_{-\pi}^{\pi/2} f(x) dx = 1$ ), calculer l’erreur  $\varepsilon$  commise dans cette méthode. Commenter.
3. Donner une valeur de  $n$  pour laquelle l’erreur commise soit de l’ordre  $10^{-5}$  (c.à.d. la précision soit plus grande).

Solution

1. Application de la méthode des rectangles

Le nombre de sous-intervalles imposé est  $n = 10$

$$\Rightarrow h = \Delta x = \frac{b - a}{n} = \frac{\pi/2 - (-\pi)}{10} \Rightarrow h = \frac{3}{20}\pi$$

On calcul alors les points de discrétisation :

$$i = 0, \dots, 10 \Rightarrow \begin{cases} x_i = a + i \times h \\ f(x_i) = \cos(x_i) \end{cases} \text{ Par exemple :}$$

$$\begin{cases} x_0 = a + 0 \times h & \Rightarrow x_0 = -\pi \\ f(x_0) = \cos(x_0) = \cos(-\pi) & \Rightarrow f(x_0) = -1 \end{cases}$$

$$\begin{cases} x_1 = a + 1 \times h = -\pi + \frac{3}{20}\pi & \Rightarrow x_1 = -\frac{17}{20}\pi \\ f(x_1) = \cos(x_1) = \cos\left(-\frac{17}{20}\pi\right) & \Rightarrow f(x_1) = -0,891 \dots \end{cases}$$

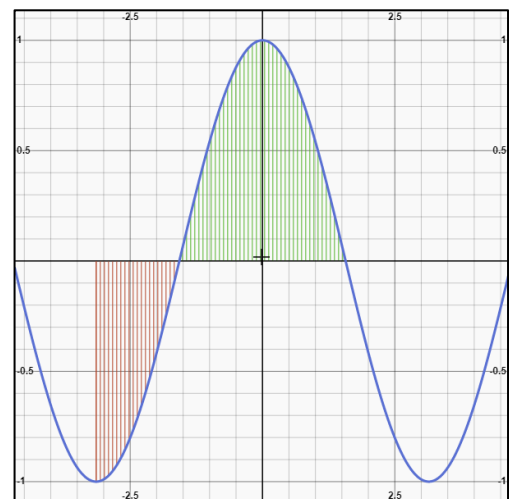


Figure 3.3 : Graphe de la fonction  $f(x) = \cos x$  schématisant l’aire estimée par l’intégrale

Ainsi de suite jusqu’à atteindre  $i = 10$ . Le tableau suivant regroupe les résultats obtenus pour chaque itération.



Tableau 3.1 : Résultats de calcul en appliquant la Méthode des Rectangles

	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$	$x_{10}$
$x_i$	$-\pi$	$-\frac{17}{20}\pi$	$-\frac{14}{20}\pi$	$-\frac{11}{20}\pi$	$-\frac{2}{5}\pi$	$-\frac{1}{4}\pi$	$-\frac{1}{10}\pi$	$\frac{1}{20}\pi$	$\frac{1}{5}\pi$	$\frac{7}{20}\pi$	$\frac{1}{2}\pi$
$f(x_i)$	-1	-0,891	-0,588	-0,156	0,309	0,707	0,951	0,988	0,809	0,454	0

On procède maintenant à la sommation (formule III.3) :

$$I_{rect} = h \times \sum_{i=0}^9 f(a + i \times h) = h \times [f(x_0) + f(x_1) + f(x_2) + \dots + f(x_9)] \Rightarrow I_{rect} = \mathbf{0,74580619}$$

**1. Calcul de l'erreur commise**

$$\varepsilon = |I - I_{rect}| = 1 - 0,74580619 \Rightarrow \varepsilon = \mathbf{0,2541981}$$

**Commenter :** On remarque que l'erreur commise est assez importante. Pour y remédier, il faudra augmenter le nombre de sous-intervalles n.

**2. Déduire le nombre n de sous-intervalles pour un  $\varepsilon_{max} \approx 10^{-5}$**

On utilise la formule de l'erreur commise (III.5) :

$$\varepsilon \leq \frac{1}{2} \frac{(b - a)^2}{n} \sup |f'(x)|_{x \in [a; b]}$$

On sait que la première dérivée de  $f(x)$  est

$$f(x) = \cos x \Leftrightarrow f'(x) = -\sin x$$

La fonction de sinus est une fonction bornée par  $M = 1$  c.à.d.

$$\sup |f'(x)| = \sup |-\sin(x)| = 1$$

On se restreint à l'intervalle  $[a, b]$  imposé :

$$\sup |-\sin(x)|_{x \in [-\pi; \pi/2]} = \mathbf{1}$$

$$\varepsilon_{max} = \frac{1}{2} \frac{\left(\frac{\pi}{2} - (-\pi)\right)^2}{n} \times 1 = 10^{-5} \Rightarrow n = \frac{1}{2} \frac{\left(\frac{\pi}{2} - (-\pi)\right)^2}{10^{-5}} = \frac{\left(\frac{\pi}{2} - (-\pi)\right)^2}{2 \times 10^{-5}} = \frac{\left(\frac{\pi}{2} - (-\pi)\right)^2}{2} \times 10^5 = k \times 10^5$$

$k$  : une constante négligeable devant  $10^5$ . Ceci implique que l'erreur maximale est aux environs de :

$$\varepsilon_{max} \approx \mathbf{10^{-5}}$$

Par conséquent, le nombre de sous-intervalles  $n$  doit être de l'ordre de  $10^5$

$$\varepsilon \propto \mathbf{1/n} \Rightarrow \mathbf{n \approx 10^5}$$

### 3.3.4 Programme de la « Méthode des Rectangles »

```

Program Rectangle
Implicit none
Integer:: i, n
doubleprecision, dimension(0:1000000):: x
doubleprecision:: s, h, a, b
real:: f
!---Entrée des Données
write(*,*) "L'intervalle d'itegration [a,b]="
read(*,*) a, b
write(*,*) "Sous-intervalles n="
read(*,*) n
h=(b-a)/n
x(0)=a
s=0
!---Calcul
do i=0, n
x(i)=x(0)+(i*h)
s=s+(f(x(i))*h)
enddo
!---Affichage du résultat
write(*,*) "le résultat de l'intégrale de f sur [a,b] est", s
End
!---La Fonction
Function f(x)
doubleprecision:: x
f=cos(x)
Return
Endfunction

```

```

"F:\Software Setups\Fort...  -  □  ×
-3.14159265
1.57079633
Sous-intervalles n=
10
le résultat de l'intégrale de f sur
[a,b] est 7.458062080406882E-001
Press any key to continue

L'intervalle d'itegration [a,b]=
-3.14159265
1.57079633
Sous-intervalles n=
100000
le résultat de l'intégrale de f sur
[a,b] est 9.999764415573808E-001
Press any key to continue

```

Figure 3.4 : Programme de la méthode des rectangles avec solution pour  $n = 10$  et  $n = 10^5$

#### Remarque

On voit bien qu'en augmentant le nombre de sous-intervalle, le résultat est d'autant plus proche de la valeur exacte  $I_{rect}(n = 10^5) \approx 0,999976 \approx 1 = I$

**N.B. :** Les fonctions trigonométriques (cosinus, sinus et tangente) sont prédéfinies en radians sur le FORTRAN, c'est pourquoi les valeurs de l'intervalle  $[a, b]$  sont entrées en radians.

## 3.4 Méthode des Trapèzes

### 3.4.1 Définition

La méthode des trapèzes, comme celle des rectangles, consiste à diviser l'intervalle  $[a, b]$  en sous-intervalles de longueurs égales. Mais pour approcher d'autant plus la courbe de la fonction  $f$ , cette méthode assimile le calcul de l'intégrale à la sommation de trapèzes très fins avec  $\Delta x$  comme base (illustré sur le schéma de la figure 3.5).

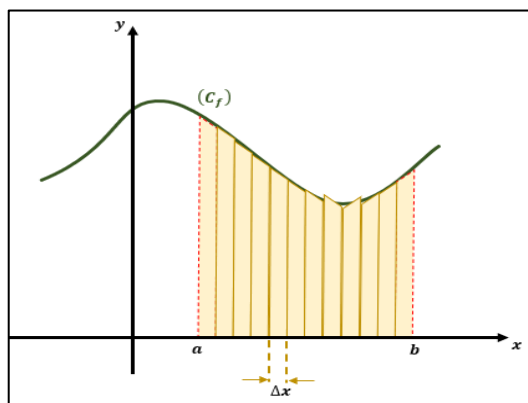


Figure 3.5 : Schéma représentant la méthode des Trapèzes

### 3.4.2 Théorème

Soit une fonction  $f$  continue sur l'intervalle  $[a, b]$  et dérivable sur l'intervalle ouvert  $]a, b[$ . On divise cet intervalle en  $x_i$  points équidistants (avec  $i = 0, \dots, n$ ). On note  $h$  le pas de cette subdivision. Prenons chaque intervalle  $[x_i ; x_{i+1}]$ , la fonction  $f$  comprise dans cet intervalle est remplacée par une fonction droite qui joint les points  $(x_i, f(x_i))$  et  $(x_{i+1}, f(x_{i+1}))$ . De ce fait, la méthode des trapèzes est une méthode d'ordre 1.

Le calcul de l'intégrale de  $f$  est déduite par :

$$\int_a^b f(x) dx \simeq \sum_{i=0}^{n-1} (x_{i+1} - x_i) \frac{f(x_i) + f(x_{i+1})}{2} \quad (\text{III. 6})$$

Avec  $(x_{i+1} - x_i) = \Delta x = h = \frac{b-a}{n}$

Lorsque la subdivision se réduit à sa plus simple expression  $x_0 = a$  et  $x_1 = b$ , on obtient :

$$\int_a^b f(x) dx \simeq \frac{1}{2} (b - a) (f(a) + f(b)) \quad (\text{III. 7})$$

L'intégrale de la fonction  $f$  sur l'intervalle  $[a, b]$  se déduit à partir de la formule suivante :

$$I_{trap} = \frac{h}{2} \left( f(a) + f(b) + \sum_{i=1}^{n-1} f(x_i) \right) \quad (\text{III. 8})$$

L'erreur  $\varepsilon$  dans la méthode des trapèzes est estimée par la formule suivante :

$$\varepsilon = |I - I_{trap}| \leq \frac{1}{12} \frac{(b-a)^3}{n^2} \sup |f''(x)|_{x \in [a,b]} \quad (\text{III. 9})$$

$I$  : intégrale de  $f$  calculée analytiquement et  $I_{trap}$  : intégrale de  $f$  calculée avec la méthode des trapèzes.

#### Remarque

On remarque bien que la précision dans la méthode des trapèzes est plus poussée que dans la méthode des rectangles car  $\varepsilon_{rect} \propto (1/n)$  et  $\varepsilon_{trap} \propto (1/n^2)$ . Ce qui veut dire que 100 sous-intervalles considérés dans la méthode des trapèzes équivaut à 10000 sous-intervalles dans celle des rectangles. Mais tout dépend aussi de la fonction et de l'amplitude de l'intervalle étudié.

### 3.4.3 L'algorithme

Les étapes de résolution par la méthode des trapèzes sont similaires à celles des rectangles :

- Le choix du nombre de discrétisation  $n$  et ainsi le pas  $h$
- L'initialisation de la somme, qui représente le résultat de l'intégrale, à  $s = \frac{h}{2} (f(a) + f(b))$
- Lancement d'une boucle allant de 1 à  $n - 1$  pour faire :
  - Le calcul des points de discrétisation  $x_i$  dans l'intervalle  $[a, b]$
  - Le produit de  $\left(\frac{h}{2}\right) \times f(x_i)$
  - Ajouter le résultat du produit à la somme  $s$

- Tester si la précision est atteinte (en utilisant la formule de  $\varepsilon$ ) : si oui => sortir de la boucle, si non => continuer la boucle
- Affichage du dernier résultat de la somme  $s$  (résultat de l'intégrale).

**Exemple**

Soit la fonction  $f$  définie sur  $\mathbb{R}$  par :  $f(x) = \sin x$

1. Calculer, à l'aide de la méthode des trapèzes, une valeur approchée de  $I$  ( $I = \int_{-\pi}^{\pi} f(x) dx$ ) en utilisant  $n = 10$  sous-intervalles.
2. Sachant que la valeur exacte de l'intégrale est ( $I = \int_{-\pi}^{\pi} f(x) dx = 0$ ), calculer l'erreur  $\varepsilon$  commise dans cette méthode. Commenter.
3. Comparer ce résultat avec celui obtenu dans la méthode précédente

**Solution**

**1. Application de la méthode des trapèzes**

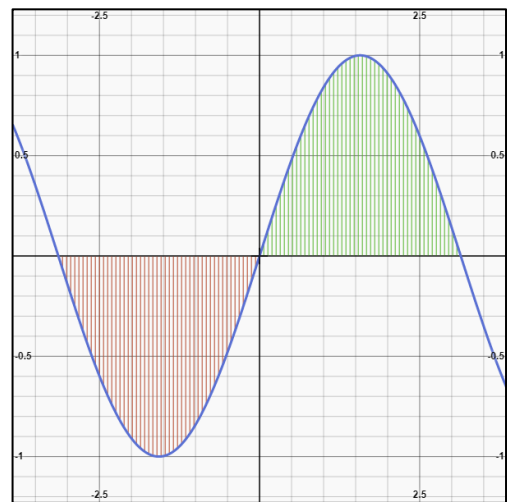
Le nombre de sous-intervalles imposé est  $n = 10$

$$\Rightarrow h = \Delta x = \frac{b - a}{n} = \frac{\pi - (-\pi)}{10} \Rightarrow h = \frac{1}{5}\pi$$

On calcul alors les points de discrétisation :

$$i = 0, \dots, 10 \Rightarrow \begin{cases} x_i = a + i \times h \\ f(x_i) = \sin(x_i) \end{cases} \text{ Par exemple :}$$

$$\begin{cases} x_0 = a + 0 \times h & \Rightarrow x_0 = -\pi \\ f(x_0) = \sin(x_0) = \sin(-\pi) & \Rightarrow f(x_0) \approx 0 \\ x_1 = a + 1 \times h = -\pi + \frac{1}{5}\pi & \Rightarrow x_1 = -\frac{4}{5}\pi \\ f(x_1) = \sin(x_1) = \sin\left(-\frac{4}{5}\pi\right) & \Rightarrow f(x_1) = -0,5878 \dots \end{cases}$$



**Figure 3.6 :** Graphe de la fonction  $f(x) = \sin x$  schématisant l'aire estimée par l'intégrale

Le tableau suivant regroupe les résultats obtenus pour chaque itération.

**Tableau 3.2 :** Résultats de calcul en appliquant la Méthode des Trapèzes

	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$	$x_{10}$
$x_i$	$-\pi$	$-\frac{4\pi}{5}$	$-\frac{3\pi}{5}$	$-\frac{2\pi}{5}$	$-\frac{\pi}{5}$	0	$\frac{\pi}{5}$	$\frac{2\pi}{5}$	$\frac{3\pi}{5}$	$\frac{4\pi}{5}$	$\pi$
$f(x_i)$	0	-0.588	-0.951	-0.951	-0.588	0	0.588	0.951	0.951	0.588	0

On procède maintenant à la sommation (formule III.6)

$$I_{trap} = \frac{h}{2} \left( f(x_0) + f(x_{10}) + \sum_{i=1}^9 f(x_i) \right) = \frac{\pi}{10} (0 + 0 + (f(x_1) + f(x_2) + \dots + f(x_9)))$$

$$\Rightarrow I_{trap} = 3,61049 \times 10^{-18}$$

**1. Calcul de l'erreur commise**

$$\varepsilon = |I - I_{trap}| = |0 - 3,61049 \times 10^{-18}| \Rightarrow \varepsilon = 3,61049 \times 10^{-18}$$

Ce qui est pratiquement zero

## 2. Comparaison entre les deux méthodes Trapèze et Rectangle

Avec un petit nombre de sous-intervalle ( $n = 10$ ), la méthode des trapèzes est arrivée à une précision assez poussée. Quant à la méthode des rectangles, il a fallu augmenter considérablement le nombre de sous-intervalle ( $n = 10^5$ ) pour atteindre une précision qui reste encore inférieure à celle des trapèzes.

De ce fait, on conclut que la méthode des trapèzes est beaucoup plus précise que la méthode des rectangles.

```

Program Trapeze
Implicit none
Integer:: i, n=10
doubleprecision, dimension(0:10):: x
doubleprecision:: s, h, a, b
real:: f
!---Entrée des Données
write(*,*) "L'intervalle d'itegration [a,b]="
read(*,*) a, b
h=(b-a)/n
x(0)=a
s=(h/2)*f(x(0))
!---Calcul
do i=1, n
x(i)=a+(i*h)
s=s+(h/2)*(f(x(i)))
enddo
!---Affichage du résultat
write(*,*) "le résultat de l'intégrale de f sur [a,b] est", s
End
!---La Fonction
Function f(x)
doubleprecision:: x
f=sin(x)
Return
Endfunction

```

*Figure 3.7 : Programme de la méthode des trapèzes avec solution pour  $n = 10$*

**N.B. :** Comme on le voit sur le résultat obtenu par programme, il n'est pas exactement égal à celui calculé manuellement. Ceci est due majoritairement à deux raisons : (1) Le Fortran effectue des arrondissements lors de ses calculs c.à.d. qu'il ne prenne pas tous les chiffres après la virgule et se résume à arrondir le nombre pour effectuer ses calculs. (2) Sur Fortran, les fonctions trigonométriques sont calculées à partir de leur développement de Taylor c.à.d. en faisant des approximations et non pas des calculs analytiques exactes. Le cumule de ses approximations et arrondissements conduit à un cumule des erreurs, ce qui fait que le résultat est légèrement différent de celui calculé manuellement.

## 3.5 Méthode de Simpson

### 3.5.1 Définition

Dans la méthode de Simpson, la fonction  $f$  est remplacée par une fonction de second ordre définissant un arc parabolique passant par les points d'ordonnées  $f(x_i)$ ;  $f(m_i)$  et  $f(x_{i+1})$  (où  $m_i$  est le milieu du sous-intervalle  $[x_i ; x_{i+1}]$ ). Le schéma de la figure III.4 illustre bien cette méthode.

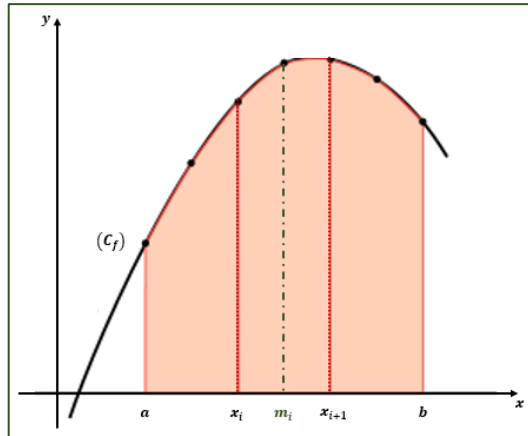


Figure 3.8 :Schéma représentant la méthode de Simpson

### 3.5.2 Théorème

Soit une fonction  $f$  continue sur l'intervalle  $[a, b]$  et dérivable sur l'intervalle ouvert  $]a, b[$ . On divise cet intervalle en  $x_i$  points équidistants (avec  $i = 0, \dots, n$ ). On note  $h$  le pas de cette subdivision. Prenons chaque intervalle  $[x_i ; x_{i+1}]$ , la fonction  $f$  dans cet intervalle est remplacée par un polynôme de second ordre qui passe par les trois points  $(x_i, f(x_i)) ; (m_i, f(m_i))$  et  $(x_{i+1}, f(x_{i+1}))$ .

L'intégrale de  $f$  est calculée par la relation suivante :

$$\int_a^b f(x)dx \simeq \sum_{i=0}^{n-1} \frac{1}{6}(x_{i+1} - x_i) \left( f(x_{i+1}) + f(x_i) + 4f\left(\frac{x_{i+1} + x_i}{2}\right) \right) \quad (\text{III. 10})$$

Lorsque la subdivision de  $[a, b]$  se réduit à sa plus simple expression  $\begin{cases} x_0 = a \\ x_1 = (a + b)/2 \\ x_2 = b \end{cases}$ , la formule précédente devient :

$$\int_a^b f(x)dx \simeq \frac{1}{3}(b - a) \sum_{i=0}^{n-1} \left( f(a) + 4f\left(\frac{a + b}{2}\right) + f(b) \right) \quad (\text{III. 11})$$

La somme finale qui permet d'évaluer l'intégrale est :

$$I_{Sim} = \frac{h}{6} \sum_{i=0}^{n-1} \left( f(a + ih) + f(a + (i + 1)h) + 4f\left(a + ih + \frac{h}{2}\right) \right) \quad (\text{III. 12})$$

L'erreur  $\varepsilon$  dans la méthode de Simpson est estimée par la formule suivante :

$$\varepsilon = |I - I_{Sim}| \leq \frac{1}{2880} \frac{(b - a)^5}{n^4} \sup_{x \in [a, b]} |f^{(5)}(x)| \quad (\text{III. 13})$$

$I$  : intégrale de  $f$  calculée analytiquement et  $I_{Sim}$  : intégrale de  $f$  calculée avec la méthode de Simpson.

### 3.5.3 L'algorithme

Les étapes de l'algorithme de Simpson sont très similaires à celles des rectangles et trapèzes. Commenant par un choix du nombre de sous-intervalles  $n$ , passant par le calcul de la somme discrétisée jusqu'à atteindre la précision voulu puis l'affichage du résultat final.

**Exemple**

Soit la fonction  $f$  définie sur  $\mathbb{R}$  par :  $f(x) = 8x^4 - 8x^2 + 1$

1. Calculer, à l'aide de la méthode de Simpson, une valeur approchée de  $I$  ( $I = \int_{-1}^1 f(x) dx$ ) en utilisant  $n = 10$  ; 100 puis 1000 sous-intervalles.
2. Sachant que la valeur exacte de l'intégrale est ( $I = \int_{-1}^1 f(x) dx = -\frac{2}{15} = -0,13333$ ), calculer l'erreur  $\varepsilon$  commise pour chaque nombre de sous-intervalles. Commenter.

**Solution**

**1. Application de la méthode de Simpson**

Premier cas :  $n = 10$

Le nombre de sous-intervalles imposé est  $n = 10$

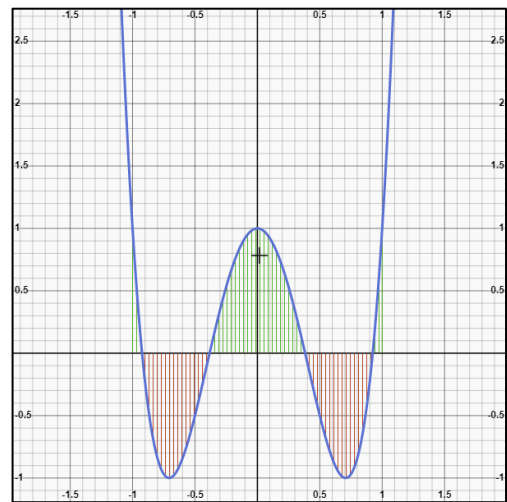
$$\Rightarrow h = \Delta x = \frac{b - a}{n} = \frac{1 - (-1)}{10} \Rightarrow h = \frac{2}{10} = \frac{1}{5}$$

On calcul alors les points de discrétisation :

$$i = 0, \dots, 10 \Rightarrow \begin{cases} x_i = a + i \times h \\ f(x_i) = \sin(x_i) \end{cases} \text{ Par exemple :}$$

$$\begin{cases} x_0 = a + 0 \times h & \Rightarrow x_0 = -1 \\ f(x_0) = 8x_0^4 - 8x_0^2 + 1 & \Rightarrow f(x_0) = 1 \end{cases}$$

$$\begin{cases} x_1 = a + 1 \times h = -1 + \frac{1}{5} & \Rightarrow x_1 = -0,8 \dots \\ f(x_1) = 8x_1^4 - 8x_1^2 + 1 & \Rightarrow f(x_1) = -0,8432 \end{cases}$$



**Figure 3.9 :** Graphe de la fonction  $f(x) = 8x^4 - 8x^2 + 1$  schématisant l'aire estimée par l'intégrale

On suit les étapes citées dans l'algorithme pour résoudre les trois cas. Pour faciliter le processus, on utilise directement le programme conçu pour la méthode de Simpson.

1.1. Programme FORTRAN de la « méthode de Simpson »

Le tableau suivant résume les résultats obtenus pour chaque nombre de sous-intervalles :

**Tableau 3.3 :** Résultats obtenus par le programme FORTRAN de la Méthode de Simpson (pour  $n = 10 ; 100 ; 1000$ )

	$n = 10$	$n = 100$	$n = 1000$
$I_{Sim}$	-0,027520026	-0.132266747	-0,133322668
$\varepsilon$	0,105813314	$1,06659 \times 10^{-3}$	$1,06721 \times 10^{-5}$

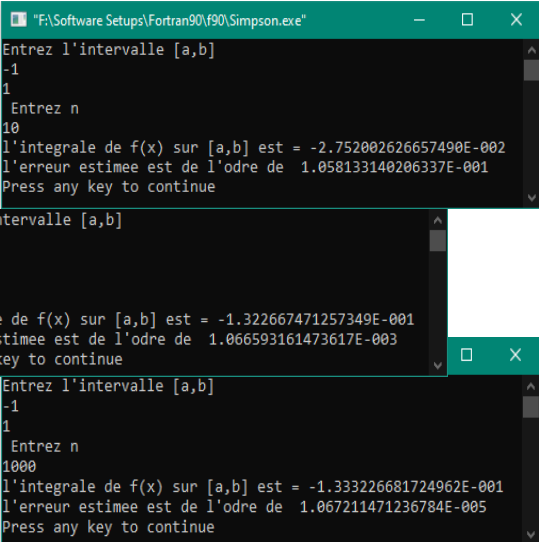
**Remarque**

On constate, d'après les résultats obtenus pour chaque  $n$ , que la valeur de l'intégrale calculée par la méthode de Simpson tend de plus en plus vers la valeur exacte de l'intégrale quand le nombre de sous-intervalles  $n$  augmente.

```

Program Simpson
Implicit none
integer::i, n
doubleprecision::s, h, a, b, int, epsilon
real:: f
doubleprecision, dimension(0:100000)::x
!---Les données
Write(*,*) "Entrez l'intervalle [a,b]"
Read(*,*) a, b
Write(*,*) " Entrez n"
Read(*,*) n
int=-0.13333333333333332
x(0)=a
h=(b-a)/n
!---Initialisation de la somme S
s=0
!---Calcul de l'intégrale
Do i=1, n
x(i)=x(0)+i*h
s=s+(h/6)*(f(x(i-1))+f(x(i))+4*f(x(i-1/2))))
enddo
!---Estimation de l'erreur commise
epsilon=abs(int-s)
!---Affichage du Résultat
write(*,*) "l'integrale de f(x) sur [a,b] est =", s
write(*,*) "l'erreur estimee est de l'ordre de", epsilon
end
!---La fonction
Function f(x)
doubleprecision::x
f=(8*x**4)-(8*x**2)+1
Return
Endfunction

```



The figure shows three sequential screenshots of a terminal window titled "F:\Software Setups\Fortran90\F90\Simpson.exe". Each screenshot shows the user entering values for 'a', 'b', and 'n', and the program outputting the integral value and the estimated error order.

- First screenshot: User enters 'a' as -1, 'b' as 1, and 'n' as 10. Output: "l'integrale de f(x) sur [a,b] est = -2.752002626657490E-002" and "l'erreur estimee est de l'ordre de 1.058133140206337E-001".
- Second screenshot: User enters 'a' as -1, 'b' as 1, and 'n' as 100. Output: "l'integrale de f(x) sur [a,b] est = -1.322667471257349E-001" and "l'erreur estimee est de l'ordre de 1.066593161473617E-003".
- Third screenshot: User enters 'a' as -1, 'b' as 1, and 'n' as 1000. Output: "l'integrale de f(x) sur [a,b] est = -1.333226681724962E-001" and "l'erreur estimee est de l'ordre de 1.067211471236784E-005".

Figure 3.10 : Programme de la méthode de Simpson montrant les résultats obtenus pour l'exemple



**Chapitre 4**  
**PROBLEMES AUX VALEURS**  
**PROPRES**

---

## Chapitre 4 PROBLEME AUX VALEURS PROPRES

### 4.1 Introduction

Le problème aux valeurs propres consiste à résoudre le système d'équations  $AX = \lambda X$  où  $\lambda$  est un paramètre qui peut prendre des valeurs complexes en générale. En termes d'équations :

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = \lambda x_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n = \lambda x_2 \\ \dots \\ a_{p1}x_1 + a_{p2}x_2 + a_{p3}x_3 + \dots + a_{pn}x_n = \lambda x_p \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n = \lambda x_n \end{cases} \tag{IV.1}$$

Pour la résolution de ce système d'équations, il faut déterminer les valeurs du paramètre  $\lambda$  appelées « **valeurs propres** » qui vérifient le système et la solution  $X^s$  appelé « **vecteur propre** » correspondant à chaque valeur.

Quand on écrit le système sous la forme équivalente en transposant le terme de droite  $\lambda x_i$  à gauche, on aura :

$$\begin{cases} (a_{11} - \lambda)x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = 0 \\ a_{21}x_1 + (a_{22} - \lambda)x_2 + a_{23}x_3 + \dots + a_{2n}x_n = 0 \\ \dots \\ a_{p1}x_1 + a_{p2}x_2 + \dots + (a_{pp} - \lambda)x_p + \dots + a_{pn}x_n = 0 \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + (a_{nn} - \lambda)x_n = 0 \end{cases} \tag{IV.2}$$

On obtient alors un système d'équations homogènes. Ecrit sous le forme matricielle  $A'X = 0$  comme suit.

$$\begin{pmatrix} \mathbf{a_{11} - \lambda} & a_{12} & a_{13} & \dots & \vdots & a_{1n} \\ a_{21} & \mathbf{a_{22} - \lambda} & a_{23} & \dots & \vdots & a_{2n} \\ \dots & \dots & \dots & \dots & \vdots & \vdots \\ a_{p1} & a_{p2} & \dots & \mathbf{a_{pp} - \lambda} & \vdots & a_{pn} \\ \dots & \dots & \dots & \dots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & \vdots & \mathbf{a_{nn} - \lambda} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{IV.3}$$

Où la matrice  $A' = A - \lambda I$  et  $I$  est la matrice identité  $N \times N$

$$\begin{pmatrix} \mathbf{a_{11} - \lambda} & a_{12} & a_{13} & \dots & \vdots & a_{1n} \\ a_{21} & \mathbf{a_{22} - \lambda} & a_{23} & \dots & \vdots & a_{2n} \\ \dots & \dots & \dots & \dots & \vdots & \vdots \\ a_{p1} & a_{p2} & \dots & \mathbf{a_{pp} - \lambda} & \vdots & a_{pn} \\ \dots & \dots & \dots & \dots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & \vdots & \mathbf{a_{nn} - \lambda} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & \vdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & \vdots & a_{2n} \\ \dots & \dots & \dots & \dots & \vdots & \vdots \\ a_{p1} & a_{p2} & \dots & a_{pp} & \vdots & a_{pn} \\ \dots & \dots & \dots & \dots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & \vdots & a_{nn} \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \vdots \\ 0 & 0 & \dots & 1 & 0 \\ \dots & \dots & \dots & \dots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix} \tag{IV.4}$$

Le système d'équations (IV. 2) est un système linéaire homogène, donc si  $\det(A') \neq 0$  on a un **système de Cramer** et la solution est **unique**  $X^s = 0$ . Par conséquent, pour avoir des solutions non nulles il faut que :

$$\det(A') = \det(A - \lambda I) = 0 \tag{IV. 5}$$

### 4.2 Méthode Polynomiale

La relation  $\det(A - \lambda I) = 0$  donne en général un polynôme de degrés n appelé « polynôme caractéristique » de la forme :

$$p(\lambda) = (-1)^n \lambda^n + P_1 \lambda^{n-1} + P_2 \lambda^{n-2} + \dots + P_{n-1} \lambda + P_n \tag{IV. 6}$$

Les valeurs propres  $\lambda$  sont les racines de ce polynôme.

#### Exemple 1 (système à 2 × 2)

Soit le problème suivant :

$$\begin{array}{l} x_1 + 2x_2 = \lambda x_1 \\ 5x_1 + 4x_2 = \lambda x_2 \end{array} \quad \begin{array}{l} \text{En notation matricielle} \\ \Rightarrow \end{array} \quad \begin{pmatrix} 1 & 2 \\ 5 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \lambda \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

Dans ce cas

$$A = \begin{pmatrix} 1 & 2 \\ 5 & 4 \end{pmatrix} \quad \text{Donc} \quad A' = A - \lambda I = \begin{pmatrix} 1 & 2 \\ 5 & 4 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 - \lambda & 2 \\ 5 & 4 - \lambda \end{pmatrix}$$

Donc le polynôme caractéristique de ce système d'après la forme générale (IV. 6) est de la forme

$$\begin{array}{l} p(\lambda) = \det(A - \lambda I) = (-1)^2 \lambda^2 + P_1 \lambda + P_2 = \lambda^2 + P_1 \lambda + P_2 \quad \text{Par identification} \\ \det(A - \lambda I) = p(\lambda) = (1 - \lambda) \times (4 - \lambda) - 10 = \lambda^2 - 5\lambda - 6 \quad \text{On a} \Rightarrow \end{array} \quad \begin{cases} P_1 = -5 \\ P_2 = -6 \end{cases}$$

Maintenant les valeurs propres s'obtiennent en calculant les racines  $p(\lambda) = 0$  qui sont  $\lambda_1 = -1$  et

$\lambda_2 = -6$ . Les vecteurs propres  $X^{(1)} = \begin{pmatrix} x_1^{(1)} \\ x_2^{(1)} \end{pmatrix}$  et  $X^{(2)} = \begin{pmatrix} x_1^{(2)} \\ x_2^{(2)} \end{pmatrix}$  associés à ces deux valeurs se calculent à partir de l'équation aux valeurs propres

$$\begin{cases} AX^{(1)} = \lambda_1 X^{(1)} \\ AX^{(2)} = \lambda_2 X^{(2)} \end{cases} \tag{IV. 7}$$

On commence par la première équation :

$$\begin{aligned} AX^{(1)} = \lambda_1 X^{(1)} &\Rightarrow \begin{cases} x_1^{(1)} + 2x_2^{(1)} = \lambda_1 x_1^{(1)} \\ 5x_1^{(1)} + 4x_2^{(1)} = \lambda_1 x_2^{(1)} \end{cases} \text{ avec } \lambda_1 = -1 \Rightarrow \begin{cases} x_1^{(1)} + 2x_2^{(1)} = -x_1^{(1)} \\ 5x_1^{(1)} + 4x_2^{(1)} = -x_2^{(1)} \end{cases} \\ &\Rightarrow x_1^{(1)} = -x_2^{(1)} \end{aligned}$$

Donc la forme générale du vecteur propre associé à  $\lambda_1$  est  $X^{(1)} = \begin{pmatrix} x_1^{(1)} \\ x_2^{(1)} \end{pmatrix} = x_1^{(1)} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$

Pour déterminer le  $x_1^{(1)}$ , on utilise alors la condition de normalisation :  $|X^{(1)}|^2 = 1 \Rightarrow x_1^{(1)} = 1/\sqrt{2}$

$$\Rightarrow X^{(1)} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Même chose pour la deuxième équation :

$$AX^{(2)} = \lambda_2 X^{(2)} \Rightarrow \begin{cases} x_1^{(2)} + 2x_2^{(2)} = \lambda_2 x_1^{(2)} \\ 5x_1^{(2)} + 4x_2^{(2)} = \lambda_2 x_2^{(2)} \end{cases} \text{ avec } \lambda_2 = 6 \Rightarrow \begin{cases} x_1^{(2)} + 2x_2^{(2)} = 6x_1^{(2)} \\ 5x_1^{(2)} + 4x_2^{(2)} = 6x_2^{(2)} \end{cases}$$

$$\Rightarrow 5x_1^{(2)} = 2x_2^{(2)} \Rightarrow x_2^{(2)} = \frac{5}{2}x_1^{(2)}$$

Donc la forme générale du vecteur propre associé à  $\lambda_1$  est  $X^{(2)} = \begin{pmatrix} x_1^{(2)} \\ x_2^{(2)} \end{pmatrix} = x_1^{(2)} \begin{pmatrix} 1 \\ 5/2 \end{pmatrix}$

Et la normalisation donne  $|X^{(2)}|^2 = 1 \Rightarrow x_1^{(2)} = 2/\sqrt{29}$

$$\Rightarrow X^{(2)} = \frac{2}{\sqrt{29}} \begin{pmatrix} 1 \\ 5 \end{pmatrix}$$

**Exemple 2 (système à 3 × 3)**

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & -1 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \lambda \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & -1 & 3 \end{pmatrix} \quad \text{Donc} \quad A' = A - \lambda I = \begin{pmatrix} 1-\lambda & 1 & 0 \\ 0 & 2-\lambda & 0 \\ 0 & -1 & 3-\lambda \end{pmatrix}$$

Pour un système 3 × 3, le polynôme caractéristique s'écrit :

$$p(\lambda) = (-1)^3 \lambda^3 + P_1 \lambda^2 + P_2 \lambda + P_3 \Rightarrow p(\lambda) = -\lambda^3 + P_1 \lambda^2 + P_2 \lambda + P_3$$

D'un autre côté

$$\det(A - \lambda I) = -\lambda^3 + 6\lambda^2 - 11\lambda + 6$$

Par identification, on obtient :

$$\begin{cases} P_1 = 6 \\ P_2 = -11 \\ P_3 = 6 \end{cases}$$

**Remarque**

Dans le cas d'un système générale  $N \times N$ , la méthode usuelle du déterminant  $\det(A - \lambda I)$  pour le calcul du polynôme caractéristique devient difficile pour « n » grand. Dans ce cas, on doit faire recours à d'autres moyens plus adaptés qui peuvent être automatisés par un programme. Parmi ces moyens de calculs, on a celle de « Fadéev – Leverrier ».

### 4.3 La Méthode de FADEEV – LEVERRIER

#### 4.3.1 Définition

La méthode de Fadéev – Leverrier permet de calculer les coefficients du polynôme caractéristique  $(P_1; P_2; P_3; \dots; P_n)$  sans utiliser la méthode du déterminant.

Elle est basée sur le calcul d'une suite de matrice par l'algorithme suivant :

$$\begin{cases} A_1 = A \\ A_k = (A_{k-1} + (-1)^n P_{k-1} I) A \end{cases} \quad k = 2 \dots n \quad \text{(IV. 8)}$$

Où  $I$  est la matrice identité  $N \times N$  en générale. On note  $B_{k-1} = A_{k-1} + (-1)^n P_{k-1} I$

Les coefficients  $P_k = (-1)^{(n+1)} \frac{tr(A_k)}{k}$  où  $tr(A_k)$  est la trace de la matrice  $A_k$  et  $n$  la dimension de la matrice.

### 4.3.2 L'algorithme

Les étapes principales de cette méthode se résume comme suit :

- a) Identifier les éléments de la matrice  $A$
- b) Calcul du premier coefficient  $P_1$
- c) Alternner les calculs entre  $A_k$  et  $P_k$  jusqu'à obtenir les  $n$  coefficients

Pour terminer la résolution de du système (résolution de l'équation linéaire), il faudra procéder à l'utilisation de l'une des méthodes déjà employée dans le chapitre 1.

#### Exemple 1

On applique cette méthode à l'exemple  $3 \times 3$  précédent :  $A = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & -1 & 3 \end{pmatrix}$

- $P_1 = (-1)^{(3+1)} \frac{tr(A_1)}{1}$  D'après l'algorithme (IV. 8) on a  $A_1 = A \Rightarrow P_1 = \frac{tr(A)}{1} = 6$
- $P_2 = (-1)^{(3+1)} \frac{tr(A_2)}{2}$  On peut calculer le  $A_2$  à partir de l'algorithme (IV. 8)

$$A_2 = (A_1 + (-1)^3 P_1 I)A = \left( \begin{pmatrix} 1 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & -1 & 3 \end{pmatrix} - 6 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \right) \begin{pmatrix} 1 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & -1 & 3 \end{pmatrix} = \begin{pmatrix} -5 & -3 & 0 \\ 0 & -8 & 0 \\ 0 & 1 & -9 \end{pmatrix}$$

$$P_2 = \frac{tr(A_2)}{2} = -11$$

- De même  $P_3 = (-1)^{(3+1)} \frac{tr(A_3)}{3}$  On doit calculer  $A_3$

$$A_3 = (A_2 + (-1)^3 P_2 I)A = \left( \begin{pmatrix} -5 & -3 & 0 \\ 0 & -8 & 0 \\ 0 & 1 & -9 \end{pmatrix} + 11 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \right) \begin{pmatrix} 1 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & -1 & 3 \end{pmatrix} = \begin{pmatrix} 6 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 6 \end{pmatrix}$$

$$P_3 = \frac{tr(A_3)}{3} = \frac{18}{3} = 6$$

Le polynôme caractéristique du système s'écrit alors comme suit :

$$p(\lambda) = -\lambda^3 + 6\lambda^2 - 11\lambda + 6$$

#### Exemple 2

Le système  $3 \times 3$  suivant :  $\begin{cases} x_1 + 2x_2 + x_3 = \lambda x_1 \\ x_1 + 3x_2 + x_3 = \lambda x_2 \\ x_1 + 2x_2 + 2x_3 = \lambda x_3 \end{cases} \Rightarrow \begin{pmatrix} 2 & 2 & 1 \\ 1 & 3 & 1 \\ 1 & 2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \lambda \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$  avec la matrice

$$A = \begin{pmatrix} 2 & 2 & 1 \\ 1 & 3 & 1 \\ 1 & 2 & 2 \end{pmatrix}$$

La forme générale du polynôme caractéristique dans ce cas est :

$$p(\lambda) = (-1)^3 \lambda^3 + P_1 \lambda^2 + P_2 \lambda + P_3 = -\lambda^3 + P_1 \lambda^2 + P_2 \lambda + P_3$$

Les coefficients  $P_k$  se détermine par la formule suivante :  $P_k = (-1)^{(n+1)} \frac{tr(A_k)}{k}$

Tel que les matrices  $A_k$  se calculent à partir l’algorithme de Fadeev-Leverrier, formule(IV. 8)

On a  $n = 3$

- $P_1 = (-1)^{(4)} \frac{tr(A_1)}{1}$  D’après l’algorithme (IV. 8) on a  $A_1 = A \Rightarrow P_1 = \frac{tr(A)}{1} = 7$
- $P_2 = (-1)^{(4)} \frac{tr(A_2)}{2}$  On peut calculer  $A_2$  à partir de l’algorithme (IV. 8)

$$A_2 = (A_1 + (-1)^3 P_1 I)A = \left( \begin{pmatrix} 2 & 2 & 1 \\ 1 & 3 & 1 \\ 1 & 2 & 2 \end{pmatrix} - 7 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \right) \begin{pmatrix} 2 & 2 & 1 \\ 1 & 3 & 1 \\ 1 & 2 & 2 \end{pmatrix} = \begin{pmatrix} -7 & -2 & -1 \\ -1 & -8 & -1 \\ -1 & -2 & -7 \end{pmatrix}$$

$$P_2 = \frac{tr(A_2)}{2} = -11$$

- De même  $P_3 = (-1)^{(3+1)} \frac{tr(A_3)}{3}$  On doit calculer  $A_3$

$$A_3 = (A_2 + (-1)^3 P_2 I)A = \left( \begin{pmatrix} -7 & -2 & -1 \\ -1 & -8 & -1 \\ -1 & -2 & -7 \end{pmatrix} + 11 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \right) \begin{pmatrix} 2 & 2 & 1 \\ 1 & 3 & 1 \\ 1 & 2 & 2 \end{pmatrix} = \begin{pmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{pmatrix}$$

$$P_3 = \frac{tr(A_3)}{3} = \frac{15}{3} = 5$$

Le polynôme caractéristique du système s’écrit alors comme suit :

$$p(\lambda) = -\lambda^3 + 7\lambda^2 - 11\lambda + 5$$

Un autre exemple

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

On suit les mêmes étapes et on obtient :

$$p(\lambda) = -\lambda^3 + 6\lambda^2 - 11\lambda + 6$$

**N.B.**

Cette méthode permet seulement de calculer l’expression du polynôme caractéristique  $p(\lambda)$  et non pas les valeurs propres qu’on calcule après en résolvant l’équation  $p(\lambda) = 0$ . Si le polynôme est de degré  $n$  grand  $n > 4$  où les méthodes d’algèbres usuelles sont inapplicables, on fait recours aux méthodes numériques comme la méthode de Newton-Raphson, de bisection ou de point fixe.

### 4.3.3 Programme de la « Méthode de Fadeev – Leverrier »

```

implicit none
integer :: i,j,k,m
integer,parameter:: n=3
double precision,dimension(n,n,n):: a,b
double precision,dimension(n):: p
!Entrée des éléments de la matrice A
do i=1,n
do j=1,n
write(*,*) ' donner l''element a('i,j,','1)''
read(*,*) a(i,j,1)
enddo
enddo
!Calcul du coefficient P1
p(1)=0.d0
do i=1,n
p(1)=p(1)+((-1)**(n+1))*a(i,i,1)
enddo
write(*,*) 'le coefficient p1 est=', p(1)

!Calcul de la suite des matrices A(k)
do k=2,n
do i=1,n
do j=1,n
if(i.eq.j)then
b(i,j,k)= a(i,j,k-1) +((-1)**n)*p(k-1)
else
b(i,j,k)=a(i,j,k-1)
endif
write(*,*) 'b', i,j,k, '=', b(i,j,k)
enddo
enddo

!Calcul du produit A(k)= B(i,j,k)*A(i,j,1)
do i=1,n
do j=1,n
a(i,j,k)=0.d0
do m=1,n
a(i,j,k)=a(i,j,k)+b(i,m,k)*a(m,j,1)
enddo
enddo
enddo
!Calcul des coefficients restants P2 et P3
p(k)=0
do i=1,n
p(k)=p(k)+((-1)**(n+1))*a(i,i,k)
enddo
p(k)=p(k)/k
write(*,*) 'le coefficient p',k,'est=', p(k)
enddo
enddo
end

```

```

Sélection "F:\Software Setups\Fortran90\f90\methode_...
le coefficient p1 est=      6.0000000000000000
le coefficient p 2est=     -11.0000000000000000
le coefficient p 3est=      6.0000000000000000
Press any key to continue

```

*Figure 4.1 : Programme de la méthode de « Fadeev – Leverrier » avec solution de l'exemple*

### Exercice 1

- a) Déterminer le polynôme caractéristique puis en déduire les valeurs propres de la matrice A en utilisant l'algorithme de Fadeev-Leverrier

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 4 \\ 0 & 1 & 0 \end{pmatrix}$$

$$\begin{cases} A_1 = A \\ A_k = (A_{k-1} + (-1)^n P_{k-1} I) A \end{cases} \quad P_m = (-1)^{n+1} \frac{\text{tr}(A_m)}{m} (*)$$

Où  $P_m$  sont les coefficients du polynôme caractéristique.

On donne :  $x^3 - x^2 - 4x + 4 = (x - 1)(x^2 - 4)$

- b) En supposant que les matrices  $A_m$  sont données, écrire un programme Fortran qui calcule les coefficients  $P_m$  avec la relation (\*) donnée ci-dessus.

**Exercice 2**

Utiliser la méthode de Fadeev-Leverrier pour obtenir le polynôme caractéristique  $p(\lambda)$  du problème aux valeurs propres suivant :

$$\begin{aligned} 2x_1 + 2x_2 + x_3 &= \lambda x_1 \\ x_1 + 3x_2 + x_3 &= \lambda x_2 \\ x_1 + 2x_2 + 2x_3 &= \lambda x_3 \end{aligned}$$

Même question pour un problème aux valeurs propres avec une matrice :

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

**4.4 Méthode de Jacobi**

**4.4.1 Définition**

La méthode de Jacobi est une méthode itérative applicable à une matrice A symétrique. Elle consiste à faire opérer le groupe des rotations planes sur A, c'est-à-dire à multiplier A par des transformations orthogonales afin de la mettre sous forme diagonale, les éléments diagonaux étant les valeurs propres de la matrice A. À la première étape, on calcule la matrice  $A_1 = H^t A H$ . En itérant ce processus, on obtient :

$$A_{k+1} = (H_1^t H_2^t \dots H_k^t) A (H_1 H_2 \dots H_k) \tag{IV. 9}$$

La suite des matrices  $A_k$  converge vers une matrice diagonale dont les éléments diagonaux sont les valeurs propres de la matrice initiale A. La suite des matrices  $P_k = H_1 H_2 \dots H_k$  converge vers la matrice dont les colonnes sont constituées de vecteurs propres.

Considérons une matrice réelle symétrique A, qu'on désire diagonaliser. Commençons par le cas très simple d'une matrice d'ordre 2.

**Exemple**

Soit une matrice 2x2  $\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$

La diagonalisation se fera via l'application d'une matrice orthogonale, dans ce cas une simple matrice de rotation dans le plan :

$$H = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}$$

$$H^{-1} A H = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$



**Remarque**

L'angle  $\theta$  est déterminé par l'annulation des composantes hors diagonale

$$-\sin(\theta) \cos(\theta) (a_{11} - a_{22}) + a_{12}(\sin(\theta)^2 - \cos(\theta)^2) = 0$$

Ce qui donne :

$$\begin{aligned} \frac{1}{2} \sin(2\theta) (a_{22} - a_{11}) &= a_{12} \cos(2\theta) \\ \Rightarrow \theta &= \frac{1}{2} \arctan\left(\frac{a_{12}}{a_{22} - a_{11}}\right) \end{aligned}$$

Dans le cas générale ( $n > 2$ ), La méthode de Jacobi consiste à appliquer une suite de transformation du type cité ci-dessus, cette fois à l'aide de matrices de rotation générale  $H(i, j, \theta)$  qui effectuent une rotation dans le plan  $x_i - x_j$  en utilisant une matrice de la forme :

$$H(i, j, \theta) = \begin{pmatrix} 1 & \dots & \dots & \dots & 0 \\ 0 & \vdots & \vdots & \vdots & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & \vdots & 0 \\ 0 & \sin(\theta) & \cos(\theta) & \vdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix} \quad (\text{IV. 10})$$

Dans ce cas, l'angle  $\theta$  est donné par :

$$\theta = \frac{1}{2} \arctan\left(\frac{a_{ij}}{a_{jj} - a_{ii}}\right) \quad (\text{IV. 11})$$

A la base de cet algorithme, deux sous-routines connues « JACCOBI » et « EIGENSTATE » ont été construites permettant ainsi de calculer les valeurs et vecteurs propres pour des matrices allant jusqu'à de grande dimension. Pour les utiliser, il suffit de les appeler à l'intérieur du programme principal avec les arguments appropriés (voir programme ci-dessous).

## 4.4.2 Programme de la « Méthode de Jacobi »

```

!Programme PRINCIPAL
implicit none
integer:: i,n,m,nrot ,j
integer, parameter:: np=1000
double precision, dimension(np):: d
double precision,dimension(np,np):: a,v

write(*,*) " donner l'ordre de la matrice A"
read(*,*) n
write(*,*) " donner les coeficients de la matrice A"
do i=1,n
do j=1,n
write(*,*) " donner le coeficient", i,j," de la matrice A"
read(*,*) a(i,j)
enddo
enddo
call jacobi(a,n,np,d,v,nrot)
call eigprt(d,v,n,np)
do j=1,n
write(*,*) "la valeur propre",j,"est:", " ", d(j)
write(*,*) d(j)
write(*,*) "son vecteur propre correspondant est :"
do i=1,n
write(*,*) v(i,j)
enddo
enddo
END

```

*Figure 4.2 : Programme PRINCIPAL de la « Méthode de Jacobi »*

```

!EIGENSTATE SUBROUTINE
SUBROUTINE eigprt(d,v,n,np)
INTEGER n,np
DOUBLE PRECISION d(np),v(np,np)

INTEGER i,j,k
DOUBLE PRECISION p
do i=1,n-1
k=i
p=d(i)
do j=i+1,n
if(d(j).ge.p)then
k=j
p=d(j)
endif
enddo
if(k.ne.i)then
d(k)=d(i)
d(i)=p
do j=1,n
p=v(j,i)
v(j,i)=v(j,k)
v(j,k)=p
enddo
endif
enddo
return
END

```

*Figure 4.3 : La subroutine de « EIGENSTATE »*

```

!JACOBI SUBROUTINE
SUBROUTINE jacobi(a,n,np,d,v,nrot)
INTEGER n,np,nrot,NMAX
DOUBLE PRECISION a(np,np),d(np),v(np,np)
PARAMETER (NMAX=2000)

DOUBLE PRECISION c,g,h,s,sm,t,tau,theta,tresh,b(NMAX),z(NMAX)
do ip=1,n !Initialize to the identity matrix.
do iq=1,n
v(ip,iq)=0.
enddo
v(ip,ip)=1.
enddo
do ip=1,n
b(ip)=a(ip,ip) !Initialize b and d to the diagonal of a.
d(ip)=b(ip)
z(ip)=0. !This vector will accumulate terms of the form tapq
!as in equation (11.1.14).
enddo
nrot=0
do i=1,50
sm=0.
do ip=1,n-1 !Sum off-diagonal elements.
do iq=ip+1,n
sm=sm+abs(a(ip,iq))
enddo
enddo
if(sm.eq.0.) then
return !The normal return, which relies
tresh=0.2*sm/n**2 !on quadratic converif(i.lt.4)then gence
else !to machine underflow on the first three sweeps.
tresh=0. !...thereafter.
endif
do ip=1,n-1
do iq=ip+1,n
g=100.*abs(a(ip,iq))
!After four sweeps, skip the rotation
!if the off-diagonal element is small.
if((i.gt.4).and.(abs(d(ip))+g.eq.abs(d(ip))).and.(abs(d(iq))+g.eq.abs(d(iq))))then
a(ip,iq)=0.
else if(abs(a(ip,iq)).gt.tresh)then
h=d(iq)-d(ip)
if(abs(h)+g.eq.abs(h))then

t=a(ip,iq)/h
else
theta=0.5*h/a(ip,iq) !Equation (11.1.10).
t=1./(abs(theta)+sqrt(1.+theta**2))
if(theta.lt.0.)t=-t
endif
c=1./sqrt(1+t**2)
s=t*c
tau=s/(1.+c)
h=t*a(ip,iq)
z(ip)=z(ip)-h
z(iq)=z(iq)+h
d(ip)=d(ip)-h
d(iq)=d(iq)+h
a(ip,iq)=0.

do j=1,ip-1 !Case of rotations 1 . j < p.
g=a(j,ip)
h=a(j,iq)
a(j,ip)=g-s*(h+g*tau)
a(j,iq)=h+s*(g-h*tau)
enddo
do j=ip+1,iq-1 !Case of rotations p < j < q.
g=a(ip,j)
h=a(j,iq)
a(ip,j)=g-s*(h+g*tau)
a(j,iq)=h+s*(g-h*tau)
enddo

```

```

do j=iq+1,n                                !Case of rotations q < j . n.
g=a(ip,j)
h=a(iq,j)
a(ip,j)=g-s*(h+g*tau)
a(iq,j)=h+s*(g-h*tau)
enddo
do j=1,n
g=v(j,ip)
h=v(j,iq)
v(j,ip)=g-s*(h+g*tau)
v(j,iq)=h+s*(g-h*tau)
enddo
nrot=nrot+1
endif
enddo
enddo

do ip=1,n
b(ip)=b(ip)+z(ip)
d(ip)=b(ip) !Update d with the sum of tapq.
z(ip)=0. !and reinitialize z.
enddo
enddo
pause 'too many iterations in jacobi'
return
END

```

*Figure 4.4 : La subroutine de « JACOBI »*

**Chapitre 5**  
**INTERPOLATION POLYNOMIALE**

---

## Chapitre 5 INTERPOLATION POLYNOMIALE

### 5.1 Interpolation – Définition

Souvent certaines informations (données expérimentales) concernant une fonction  $y = f(x)$  – fonction d'une variable  $x$  – sont données sous forme de tableau de valeurs discrète  $(x_i ; y_i = f(x_i))$

$x_0$	$f(x_0)$
$x_1$	$f(x_1)$
$x_2$	$f(x_2)$
$\vdots$	$\vdots$
$\vdots$	$\vdots$
$x_n$	$f(x_n)$

Dans de tel cas à chaque valeur  $x_i$  correspond une valeur  $y_i = f(x_i)$ . mais pour les valeurs  $x$  intermédiaire  $x_i < x < x_{i+1}$  les valeurs  $y = f(x)$  ne sont pas connues et on ne peut pas les calculer car la forme analytique de la fonction  $f$  est inconnue aussi.

L'interpolation est la procédure qui permet d'obtenir les valeurs intermédiaires  $y = f(x)$  correspondantes à  $x_i < x < x_{i+1}$  à partir d'un ensemble discret de point  $(x_i, f(x_i))$ . Il est impossible de trouver la forme exacte de  $f$  à partir des points  $(x_i ; f(x_i)) ; i = 1, \dots, n$ . Pour cela dans le processus d'interpolation la fonction  $f$  est remplacée par une autre fonction  $g$  qui coïncide avec  $f$  aux points  $x_i$  tel que

$$\forall i = 1, 2, \dots, n \mid f(x_i) = g(x_i) \quad (\text{V. 1})$$

Pour les valeurs intermédiaires de  $x \in x_i < x < x_{i+1}$  le calcul des valeurs  $f(x)$  est approché par les valeurs de la fonction  $g(x)$  après sa construction, puisque  $f(x) \approx g(x)$ .

$g$  est appelée « **fonction d'interpolation** » et l'ensemble des points  $(x_i, f(x_i))$  les « **points d'interpolation** ». Il y'a plusieurs types d'interpolation selon le type de la fonction  $g$  (polynomiale trigonométrique, etc.).

### 5.2 Interpolation polynomiale

Dans ce cas la fonction d'interpolation est de type est un polynôme de degrés  $n$  en générale. Si on a un ensemble de  $n + 1$  points  $(x_i, f(x_i))$ , il suffit de construire un polynôme  $p_n(x)$  de degré  $n$  dont la courbe passe par les points d'interpolation tel que :

$$p_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n \quad (\text{V. 2})$$

### 5.3 Corollaire

Pour  $n + 1$  points d'interpolation on ne peut faire correspondre qu'un seul polynôme degré  $n$ .

### 5.4 Polynôme de LAGRANGE

On commence par construire des polynômes de degré  $n \mid L_i(x)$  avec  $i = 1, 2, \dots, n$  telque:

$$L_i(x_j) = \delta_{ij} = \begin{cases} 1 & \text{pour } i = j \\ 0 & \text{pour } i \neq j \end{cases} \quad (\text{V. 3})$$

### 5.4.1 L'expression

$$p_n(x) = \sum_{i=1}^n f(x_i) L_i(x) \quad (\text{V. 4})$$

C'est un polynôme de degré  $n$  en plus  $L_i(x_i) = f(x_i)$ .

Nous avons deux conditions sur  $L_i(x_j)$  :

- " $L_i(x_j) = 0$  pour  $\forall i \neq j$ "  $\Rightarrow$  indique que  $x_j$  est une racine du polynôme  $L_i(x)$  d'où

$$L_i(x) \sim (x - x_0)(x - x_1)(x - x_2) \dots (x - x_{n-1})(x - x_n) \quad (\text{V. 5})$$

- " $L_i(x_i) = 1$ "  $\Rightarrow$  montre que la formule finale de  $L_i(x)$  est

$$L_i(x) = \frac{(x - x_0)(x - x_1)(x - x_2) \dots (x - x_{n-1})(x - x_n)}{(x_i - x_0)(x_i - x_1)(x_i - x_2) \dots (x_i - x_{n-1})(x_i - x_n)} \quad (\text{V. 6})$$

### 5.4.2 Théorème

Etant donné  $n + 1$  points d'interpolation  $(x_i, f(x_i))$ , l'unique polynôme d'interpolation de degré  $n$  passant par tous les points peut s'écrire

$$p_n(x) = \sum_{i=1}^n f(x_i) L_i(x) \quad (\text{V. 7})$$

$p_n(x)$  s'appelle polynôme de Lagrange

## 5.5 Polynôme d'Interpolation de Newton (Interpolation de Newton)

Au lieu de la forme usuelle de  $p_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ , on écrit le polynôme sous forme différente

$$p_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + a_3(x - x_0)(x - x_1)(x - x_2) + \dots + a_n(x - x_0)(x - x_1)(x - x_2) \dots (x - x_{n-1}) \quad (\text{V. 8})$$

On doit avoir  $p_n(x_i) = f(x_i)$ . A partir de cette condition, on obtient les données suivantes :

$$\begin{aligned} a_0 &= f(x_0) \\ a_0 + a_1(x_1 - x_0) &= f(x_1) \\ a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) &= f(x_2) \\ &\vdots \\ a_0 + a_1(x_n - x_0) + a_2(x_n - x_0)(x_n - x_1) &+ a_n(x_n - x_0)(x_n - x_1) \dots (x_n - x_{n-1}) = f(x_n) \end{aligned} \quad (\text{V. 9})$$

Pour déterminer le polynôme d'interpolation de Newton il faut calculer les coefficients  $a_i$  à partir des relations (V.9). On a :

$$\begin{aligned} a_0 &= f(x_0) \\ a_1 &= \frac{f(x_1) - f(x_0)}{(x_1 - x_0)} \end{aligned}$$

On note :

$$\frac{f(x_1) - f(x_0)}{(x_1 - x_0)} = f[x_0, x_1] \quad (\text{V. 10})$$

Donc  $a_1 = f[x_0, x_1]$  où  $f[x_0, x_1]$  est la première différence divisée. On montre de même que

$a_2 = f[x_0, x_1, x_2]$  où :

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{(x_1 - x_0)} \quad (\text{V. 11})$$

$f[x_0, x_1, x_2]$  est la seconde différence divisée. De même pour un coefficient  $a_i$  quelconque

$a_i = f[x_0, x_1, x_2 \dots x_i]$  où :

$$f[x_0, x_1, x_2 \dots x_i] = \frac{f[x_1, x_2, \dots x_i] - f[x_0, x_1, \dots x_{i-1}]}{(x_i - x_0)} \quad (\text{V. 12})$$

$f[x_0, x_1, x_2 \dots x_i]$  est la  $i^{\text{ème}}$  différence divisée.

## 5.6 Méthode des Coefficients Indéterminés

Soit le polynôme d'interpolation :  $p_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$  puisque la fonction coïncide avec le polynôme aux points  $x_i$  on a  $p_n(x_i) = f(x_i)$  avec  $i = 0 \dots n$  d'où

$$\begin{aligned} p_n(x_0) &= f(x_0) \\ p_n(x_1) &= f(x_1) \\ &\vdots \\ p_n(x_n) &= f(x_n) \end{aligned} \quad (\text{V. 13})$$

Ce qui donne

$$\begin{aligned} a_0 + a_1x_0 + a_2x_0^2 + a_3x_0^3 + \dots + a_nx_0^n &= f(x_0) \\ a_0 + a_1x_1 + a_2x_1^2 + a_3x_1^3 + \dots + a_nx_1^n &= f(x_1) \\ &\vdots \\ a_0 + a_1x_n + a_2x_n^2 + a_3x_n^3 + \dots + a_nx_n^n &= f(x_n) \end{aligned} \quad (\text{V. 14})$$

On obtient un système linéaire de  $n + 1$  équations dont les inconnus sont les coefficients  $a_i$ . On peut écrire ce système sous forme matricielle.

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_p & x_p^2 & \dots & x_p^n \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_p \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_p) \\ \vdots \\ f(x_n) \end{pmatrix} \quad (\text{V. 15})$$

$\begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_p & x_p^2 & \dots & x_p^n \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{pmatrix}$  est appelée la matrice de « **VONDERMONDE** »

Pour obtenir les coefficients  $a_i$  il faut résoudre le système d'équations linéaires en utilisant l'une des méthodes utilisées.



### Exemple 1 (Lagrange)

Sachant que la forme générale du polynôme d'interpolation de Lagrange est

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

tel que  $L_i(x_j) = \delta_{ij}$  où  $(x_i, f(x_i)) \forall i = 0 \dots n$  sont les points d'interpolation

Montrer que

$$L_i(x) = \frac{(x-x_0)(x-x_1)(x-x_2) \dots (x-x_{i-1})(x-x_{i+1}) \dots (x-x_n)}{(x_i-x_0)(x_i-x_1)(x_i-x_2) \dots (x_i-x_{i-1})(x_i-x_{i+1}) \dots (x_i-x_n)}$$

Calcul le polynôme d'interpolation de Lagrange  $P_n(x)$  pour les points  $(1, 2); (2, 5)$ .

### Solution

- **Montrer la formule de  $L_i(x)$**  : On démarre de la condition suivante

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

On a pour  $i \neq j \Rightarrow x_j$  est une racine pour  $L_i(x)$  ce qui implique

$$L_i(x) \sim (x-x_0)(x-x_1) \dots \underbrace{\dots (x-x_j) \dots}_{0} (x-x_n) = 0$$

Et pour  $i = j$  implique que  $L_i(x_i)$  ait la forme suivante

$$L_i(x_i) \sim \frac{(x_i-x_0)(x_i-x_1) \dots (x_i-x_j) \dots (x_i-x_n)}{(x_i-x_0)(x_i-x_1) \dots (x_i-x_j) \dots (x_i-x_n)} = 1$$

En généralisant la formule, on obtient que

$$L_i(x) = \frac{(x-x_0)(x-x_1)(x-x_2) \dots (x-x_{i-1})(x-x_{i+1}) \dots (x-x_n)}{(x_i-x_0)(x_i-x_1)(x_i-x_2) \dots (x_i-x_{i-1})(x_i-x_{i+1}) \dots (x_i-x_n)}$$

- **Calcul du polynôme d'interpolation de Lagrange  $P_n(x)$**

On a les deux points  $(x_i; f(x_i)) : \{(1, 2); (2, 5)\}$ . La forme générale du polynôme devient dans ce cas :

$$P_1(x) = \sum_{i=0}^1 f(x_i)L_i(x) = f(x_0)L_0(x) + f(x_1)L_1(x)$$

On détermine les  $L_i(x)$  :

$$L_0(x) = \frac{(x-x_0)(x-x_1)}{(x_0-x_1)} = \frac{(x-1)(x-2)}{(1-2)} \Rightarrow L_0(x) = -x^2 + 3x - 2$$
$$L_1(x) = \frac{(x-x_0)(x-x_1)}{(x_1-x_0)} = \frac{(x-1)(x-2)}{(2-1)} \Rightarrow L_1(x) = x^2 - 3x + 2$$

On déduit la formule du polynôme  $P_1(x)$

$$P_1(x) = 2[-x^2 + 3x - 2] + 5[x^2 - 3x + 2] \Rightarrow P_1(x) = 3x^2 + 3x + 6$$

---

**Exemple 2 (Newton)**

Soit une fonction  $f : [0, 6] \rightarrow \mathbb{R}$  continument différentiable sur cet intervalle. Trouver son polynôme de Newton sachant que la forme générale du polynôme d'interpolation de Newton est

$$p_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + a_3(x - x_0)(x - x_1)(x - x_2) + \dots \dots \\ + a_n(x - x_0)(x - x_1)(x - x_2) \dots (x - x_{n-1})$$

# **METHODES DES ELEMENTS FINIS**

---

## Chapitre 6 METHODES DES ELEMENTS FINIS

### 6.1 Introduction

Les origines de la méthode des éléments finis remontent aux années 1950 lorsque des ingénieurs l'utilisèrent afin de simuler des problèmes de mécanique des milieux continus déformables. Depuis, le champ d'applications s'est considérablement étendu et les fondements théoriques de la méthode se sont amplement consolidés. Depuis son introduction au milieu du XX<sup>ème</sup> siècle, cette méthode est devenue l'outil de base dans la résolution des équations aux dérivées partielles qui interviennent dans les études scientifiques ou techniques. Conçue initialement comme un procédé de calcul en mécanique des structures, c'est sa formalisation qui a permis de l'étendre efficacement à des domaines complètement différents comme la mécanique des fluides ou l'électromagnétisme.

L'objectif de ce chapitre est d'introduire les notions de base de résolution d'un certain type d'équations différentielles ordinaires par la méthode des éléments finis. Cette méthode repose sur 3 ingrédients principaux.

- a- Exprimer le problème sous forme variationnelle ou intégrale (rendre le problème faible)
- b- Discrétiser le domaine (maillage) en le divisant en sous domaines par exemple un intervalle est divisé en sous intervalles
- c- Approcher la solution  $y(x)$  de notre problème par une combinaison linéaire de fonctions de forme choisies  $\varphi_i(x)$  :

$$y(x) = \sum_{i=1}^n y_i \varphi_i(x) \quad (\text{VI. 1})$$

Les fonctions de formes  $\varphi_i(x)$  étant connues, la détermination de la solution revient donc à calculer les coefficients  $y_i$

#### Exemple

Soit le problème suivant :  $\forall x \in [a, b]$

$$-y''(x) + C(x)y(x) = f(x) \quad (\text{VI. 2})$$

Et les conditions aux limites  $\begin{cases} y(a) = 0 \\ y(b) = 0 \end{cases}$

Cette équation décrit pas mal de problèmes et de phénomènes physiques. On procède à la première étape :

### 6.2 Exprimer le problème sous forme variationnelle ou intégrale

On multiplie l'équation (VI.2) par une fonction  $v(x)$  appelé « **fonction test** » dont les conditions aux limites sont  $v(a) = v(b) = 0$ . On aura :

$$-y''(x)v(x) + C(x)y(x)v(x) = f(x)v(x) \quad (\text{VI. 3})$$

On utilise la dérivée du produit pour modifier le terme  $y''(x)v(x)$ . On a :

$$[y'(x)v(x)]' = y''(x)v(x) + y'(x)v'(x) \Leftrightarrow y''(x)v(x) = [y'(x)v(x)]' - y'(x)v'(x)$$

On remplace alors dans l'équation (VI.3) :

$$y'(x)v(x)' - [y'(x)v(x)]' + C(x)y(x)v(x) = f(x)v(x) \quad (\text{VI. 4})$$

Maintenant on intègre cette dernière équation de part et d'autre sur l'intervalle [a, b] :

$$\int_a^b y'(x)v(x)' dx - \int_a^b [y'(x)v(x)]' dx + \int_a^b C(x)y(x)v(x)dx = \int_a^b f(x)v(x)dx \quad (\text{VI. 5})$$

Avec

$$\int_a^b [y'(x)v(x)]' dx = [y'(x)v(x)]_a^b = [y'(b)v(b) - y'(a)v(a)] = 0$$

Car  $v(b) = v(a) = 0$ . Il nous reste alors :

$$\int_a^b y'(x)v(x)' dx + \int_a^b C(x)y(x)v(x)dx = \int_a^b f(x)v(x)dx \quad (\text{VI. 6})$$

### 6.3 Maillage ou discrétisation

On divise l'intervalle [a, b] en sous intervalles en choisissant un ensemble de point à l'intérieur de l'intervalle  $x_0 = a, x_1, x_2, \dots, x_i, \dots, x_n, x_{n+1} = b$  ces points sont les nœuds de la discrétisation et les sous intervalles  $[x_0, x_1], [x_1, x_2] \dots [x_i, x_{i+1}] \dots [x_n, x_{n+1}]$  sont appelés les éléments géométriques d'où le nom méthode éléments finis.

### 6.4 Choix des fonctions de formes

On construit des fonctions linéairement indépendantes  $\varphi_1(x), \varphi_2(x), \varphi_3(x) \dots \varphi_n(x)$  et on approche la solution  $y(x)$  par la combinaison (formule VI.1) :

$$y(x) = \sum_{j=1}^n y_j \varphi_j(x)$$

En plus dans chaque sous intervalle ou élément on prends  $v(x) = \varphi_i(x) \Rightarrow v'(x) = \varphi_i'(x)$

On aura n équations de la forme (VI.6) :

$$\int_a^b \sum_{j=1}^n y_j \varphi_j'(x) \varphi_i'(x) dx + \int_a^b C(x) \left[ \sum_{j=1}^n y_j \varphi_j(x) \right] \varphi_i(x) dx = \int_a^b f(x) \varphi_i(x) dx \quad (\text{VI. 7})$$

$$\sum_{j=1}^n y_j \int_a^b \varphi_j'(x) \varphi_i'(x) dx + \sum_{j=1}^n y_j \int_a^b C(x) \varphi_j(x) \varphi_i(x) dx = \int_a^b f(x) \varphi_i(x) dx \quad (\text{VI. 8})$$

Maintenant on pose :

$$\int_a^b \varphi_j'(x) \varphi_i'(x) dx = M_{ij}$$

$$\int_a^b C(x) \varphi_j(x) \varphi_i(x) dx = P_{ij} \quad (\text{VI. 9})$$

$$\int_a^b f(x)\varphi_i(x)dx = F_{ij}$$

En remplaçant dans l'équation précédente, on aura :

$$\sum_{j=1}^n M_{ij}y_j + \sum_{j=1}^n P_{ij}y_j = \sum_{j=1}^n (M_{ij} + P_{ij})y_j = F_i \tag{VI. 10}$$

En posant  $(M_{ij} + P_{ij}) = A_{ij}$  :

$$\sum_{j=1}^n A_{ij}y_j = F_i \quad ; \quad i = 1, \dots, n \tag{VI. 11}$$

Si on écrit explicitement ces équations pour chaque  $i$  :

$$\begin{aligned} A_{11}y_1 + A_{12}y_2 + A_{13}y_3 + \dots + A_{1n}y_n &= F_1 \\ A_{21}y_1 + A_{22}y_2 + A_{23}y_3 + \dots + A_{2n}y_n &= F_2 \\ &\dots \\ A_{n1}y_1 + A_{n2}y_2 + A_{n3}y_3 + \dots + A_{nn}y_n &= F_n \end{aligned} \tag{VI. 12}$$

Donc on obtient un système de  $n$  équations à  $n$  inconnues (les  $y_i$ ) qui, sous forme matricielle, s'écrivent :

$$\begin{pmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ A_{21} & A_{22} & \dots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \dots & A_{nn} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} F_1 \\ F_2 \\ \vdots \\ F_n \end{pmatrix} \equiv AY = F \tag{VI. 13}$$

La résolution de ce système linéaire nous permettra de calculer les inconnues  $y_i (i = 1 \dots n)$  et d'avoir par la suite l'expression approchée de la solution  $y(x)$  (Formule VI.1)  $\Rightarrow y(x) = \sum_{j=1}^n y_j \varphi_j(x)$

Puisque les fonctions de formes sont données la solution est déterminé dès que les  $y_j$  calculées en résolvant le système, pour cela on a besoin de calculer les coefficients de la matrice A et celle de F à partir des intégrales définies ci-dessus

- $A_{ij} = (M_{ij} + P_{ij}) = \int_a^b \varphi_j'(x)\varphi_i'(x)dx + \int_a^b C(x)\varphi_j(x)\varphi_i(x)dx$
- $F_i = \int_a^b f(x)\varphi_i(x)dx$

(VI. 14)

**Exemple d'application**

Soit le problème suivant :

$$y''(x) + y(x) = x$$

Tel que  $x \in [0, 1]$  et dont les conditions aux limites sont  $\begin{cases} y(0) = 0 \\ y(1) = 0 \end{cases}$

On met l'équation sous la forme (VI.2) en multipliant par -1 on aura

$$-y''(x) - y(x) = -x$$

Tel que  $x \in [0, 1]$  et dont les conditions aux limites sont  $\begin{cases} y(0) = 0 \\ y(1) = 0 \end{cases}$

**a. le maillage ou discrétisation de l'intervalle :**

A titre d'exemple, on choisit seulement 2 points dans l'intervalle  $[0, 1]$  (c.à.d.  $n = 2$ ) pour faciliter les calculs. En réalité le nombre  $n$  doit être plus grand pour obtenir une bonne approximation de notre solution.

$$0 \text{ --- } x_1 \text{ --- } x_2 \text{ --- } 1 \quad \Leftrightarrow \quad x_0 = 0; x_1 = \frac{1}{3}; x_2 = \frac{2}{3}; x_3 = 1$$

Par conséquent on a 3 éléments formant les sous intervalles  $[x_0, x_1]$ ;  $[x_1, x_2]$  et  $[x_2, x_3]$

**b. Fonctions de formes :**

On se donne les fonctions de formes suivantes

$$\varphi_i(x) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}} & \text{si } x_{i-1} < x \leq x_i \\ \frac{x - x_i}{x_{i+1} - x_i} & \text{si } x_i < x \leq x_{i+1} \\ 0 & \text{si } x \leq x_{i-1} \quad \text{ou} \quad x_{i+1} < x \end{cases} \quad (\text{VI. 15})$$

Le nombre de points étant  $n = 2$ , le système linéaire final est  $2 \times 2$  et il s'écrit donc comme suit :

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} \quad (\text{VI. 16})$$

Le but est de déterminer  $y_1$  et  $y_2$  pour cela on doit calculer les  $A_{ij}$  et  $F_i$  par les relations (VI.14).

Dans notre exemple  $C(x) = -1$  et  $f(x) = -x$

$$A_{11} = (M_{11} + P_{11}) = \int_0^1 \varphi_1'(x)\varphi_1'(x)dx + \int_0^1 (-1)\varphi_1(x)\varphi_1(x)dx$$

$$\varphi_1(x) = \begin{cases} \frac{x - x_0}{x_1 - x_0} & \text{si } x_0 < x \leq x_1 \\ \frac{x - x_1}{x_2 - x_1} & \text{si } x_1 < x \leq x_2 \\ 0 & \text{si } x \leq x_0 \quad \text{ou} \quad x_2 < x \end{cases}$$

Avec  $x_0 = 0$ ;  $x_1 = \frac{1}{3}$ ;  $x_2 = \frac{2}{3}$ ;  $x_3 = 1$

$$\Rightarrow \varphi_1(x) = \begin{cases} 3x & \text{si } 0 < x \leq \frac{1}{3} \\ 3x - 1 & \text{si } \frac{1}{3} < x \leq \frac{2}{3} \\ 0 & \text{si } x \leq 0 \quad \text{ou} \quad \frac{2}{3} < x \end{cases}$$

Maintenant on a aussi besoin de  $\varphi_1'(x)$ . D'une manière générale, d'après l'expression de  $\varphi_i(x)$ , sa dérivée est :

$$\varphi_1'(x) = \begin{cases} \frac{1}{x_i - x_{i-1}} & \text{si } x_0 < x \leq x_1 \\ \frac{1}{x_{i+1} - x_i} & \text{si } x_1 < x \leq x_2 \\ 0 & \text{si } x \leq x_0 \quad \text{ou} \quad x_2 < x \end{cases}$$

Donc

$$\Rightarrow \varphi_1'(x) = \begin{cases} \frac{1}{x_1 - x_0} = 3 & \text{si } 0 < x \leq \frac{1}{3} \\ \frac{1}{x_2 - x_1} = 3 & \text{si } \frac{1}{3} < x \leq \frac{2}{3} \\ 0 & \text{si } x \leq 0 \quad \text{ou} \quad \frac{2}{3} < x \end{cases}$$

Ou on peut dériver directement  $\varphi_1(x)$

On utilise la relation de « Chasles » pour calculer les intégrales en les décomposant sur les sous-intervalles

$$\int_0^1 = \int_0^{\frac{1}{3}} + \int_{\frac{1}{3}}^{\frac{2}{3}} + \int_{\frac{2}{3}}^1 \tag{VI. 17}$$

Mais on remarque que dans l'intervalle  $[\frac{2}{3}, 1]$   $\Rightarrow \varphi_1(x) = 0$  et  $\varphi_1'(x) = 0$

$$A_{11} = \int_0^{\frac{1}{3}} \varphi_1'(x)\varphi_1'(x)dx + \int_{\frac{1}{3}}^{\frac{2}{3}} \varphi_1'(x)\varphi_1'(x)dx + \int_0^{\frac{1}{3}} (-1)\varphi_1(x)\varphi_1(x)dx + \int_{\frac{1}{3}}^{\frac{2}{3}} (-1)\varphi_1(x)\varphi_1(x)dx$$

En utilisant les données obtenues suivantes :

Intervalle	$\varphi_1(x)$	$\varphi_1'(x)$
$[0, \frac{1}{3}]$	$3x$	$3$
$[\frac{1}{3}, \frac{2}{3}]$	$3x - 1$	$3$
$[\frac{2}{3}, 1]$	$0$	$0$

$$A_{11} = \int_0^{\frac{1}{3}} 9 dx + \int_{\frac{1}{3}}^{\frac{2}{3}} 9 dx - \int_0^{\frac{1}{3}} 9x^2 dx - \int_{\frac{1}{3}}^{\frac{2}{3}} (3x - 1)^2 dx = \frac{52}{9}$$

De même :

$$A_{22} = (M_{22} + P_{22}) = \int_0^1 \varphi_2'(x)\varphi_2'(x) dx + \int_0^1 (-1)\varphi_2(x)\varphi_2(x)dx$$

$$\varphi_2(x) = \begin{cases} \frac{x - x_1}{x_2 - x_1} & \text{si } x_1 < x \leq x_2 \\ \frac{x - x_2}{x_3 - x_2} & \text{si } x_2 < x \leq x_3 \\ 0 & \text{si } x \leq x_1 \quad \text{ou} \quad x_3 < x \end{cases}$$



$$\Rightarrow \varphi_2(x) = \begin{cases} 3x - 1 & \text{si } \frac{1}{3} < x \leq \frac{2}{3} \\ 3x - 2 & \text{si } \frac{2}{3} < x \leq 1 \\ 0 & \text{si } x \leq \frac{1}{3} \quad \text{ou} \quad 1 < x \end{cases}$$

$$\Rightarrow \varphi_2'(x) = \begin{cases} 3 & \text{si } \frac{1}{3} < x \leq \frac{2}{3} \\ 3 & \text{si } \frac{2}{3} < x \leq 1 \\ 0 & \text{si } x \leq \frac{1}{3} \quad \text{ou} \quad 1 < x \end{cases}$$

Dans l'intervalle  $\left[\frac{2}{3}, 1\right] \Rightarrow \varphi_2(x) = 0$  et  $\varphi_2'(x) = 0$

$$A_{22} = \int_{\frac{1}{3}}^{\frac{2}{3}} \varphi_2'(x)\varphi_2'(x)dx + \int_{\frac{2}{3}}^1 \varphi_2'(x)\varphi_2'(x)dx + \int_{\frac{1}{3}}^{\frac{2}{3}} (-1)\varphi_2(x)\varphi_2(x)dx + \int_{\frac{2}{3}}^1 (-1)\varphi_2(x)\varphi_2(x)dx$$

$$A_{22} = \int_{\frac{1}{3}}^{\frac{2}{3}} 9 dx + \int_{\frac{2}{3}}^1 9 dx - \int_{\frac{1}{3}}^{\frac{2}{3}} (3x - 1)^2 dx - \int_{\frac{2}{3}}^1 (3x - 2)^2 dx = \frac{52}{9}$$

De même pour  $A_{12}$  et  $A_{21}$  :

$$A_{12} = A_{21} = (M_{12} + P_{12}) = \int_0^1 \varphi_1'(x)\varphi_2'(x)dx + \int_0^1 (-1)\varphi_1(x)\varphi_2(x)dx$$

$$A_{12} = A_{21} = \int_{\frac{1}{3}}^{\frac{2}{3}} \varphi_1'(x)\varphi_2'(x)dx + \int_{\frac{1}{3}}^{\frac{2}{3}} (-1)\varphi_1(x)\varphi_2(x)dx$$

$$A_{12} = A_{21} = \int_{\frac{1}{3}}^{\frac{2}{3}} 9 dx - \int_{\frac{1}{3}}^{\frac{2}{3}} (3x - 1)^2 dx = \frac{26}{9}$$

Maintenant les  $F_i$  :

$$F_1 = \int_0^1 (-x)\varphi_1(x)dx = \int_0^{\frac{1}{3}} (-x)\varphi_1(x)dx + \int_{\frac{1}{3}}^{\frac{2}{3}} (-x)\varphi_1(x)dx$$

$$F_1 = \int_0^{\frac{1}{3}} (-x)(3x)dx + \int_{\frac{1}{3}}^{\frac{2}{3}} (-x)(3x - 1)dx = -\frac{7}{54}$$

$$F_2 = \int_0^1 (-x)\varphi_2(x)dx = \int_{\frac{1}{3}}^{\frac{2}{3}} (-x)\varphi_2(x)dx + \int_{\frac{2}{3}}^1 (-x)\varphi_2(x)dx$$

$$F_2 = \int_{\frac{1}{3}}^{\frac{2}{3}} (-x)(3x - 1)dx + \int_{\frac{2}{3}}^1 (-x)(3x - 2)dx = -\frac{13}{54}$$

Le système devient :

$$\begin{pmatrix} \frac{52}{9} & \frac{26}{9} \\ \frac{26}{9} & \frac{52}{9} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} -\frac{7}{54} \\ \frac{13}{54} \end{pmatrix}$$

$$y_1 = \frac{\begin{vmatrix} -\frac{7}{54} & \frac{26}{9} \\ -\frac{13}{54} & \frac{52}{9} \end{vmatrix}}{\begin{vmatrix} \frac{52}{9} & \frac{26}{9} \\ \frac{26}{9} & \frac{52}{9} \end{vmatrix}} = -\frac{1}{468}$$

$$y_2 = \frac{\begin{vmatrix} \frac{52}{9} & -\frac{7}{54} \\ \frac{26}{9} & -\frac{13}{54} \end{vmatrix}}{\begin{vmatrix} \frac{52}{9} & \frac{26}{9} \\ \frac{26}{9} & \frac{52}{9} \end{vmatrix}} = -\frac{19}{468}$$

La solution est :

$$y(x) = \sum_{j=1}^2 y_j \varphi_j(x) = y_1 \varphi_1(x) + y_2 \varphi_2(x)$$

$$y(x) = \begin{cases} y_1 * 3x & \text{si } 0 < x \leq \frac{1}{3} \\ y_1 * (3x - 1) + y_2 * (3x - 1) & \text{si } \frac{1}{3} < x \leq \frac{2}{3} \\ y_2 * (3x - 2) & \text{si } \frac{2}{3} < x \leq 1 \end{cases}$$

**Exercice 1**

Avec la méthode des éléments finis (Type Galerkin) transformer en un problème faible le problème :

$$-y'' + xy + x^2 = 0$$

Si on exprime  $y(x)$  sous la forme d'une combinaison linéaire de fonction de formes  $\varphi_i(x)$ , montrez qu'on peut l'écrire sous la forme matricielle  $AY = F$ . Donnez l'expression des coefficients  $A_{ij}$  et  $F_i$  en fonction des fonctions de formes  $\varphi_i(x)$ .

**Exercice 2**

Avec la méthode des éléments finis (Type Galerkin) transformer le problème suivant :  $y'' = x$  en un problème faible puis effectuer sa résolution pour  $x \in [0, 3]$  ;  $y(0) = y(3) = 0$  et les fonctions de formes  $\varphi_i(x)$  qui sont données par :

$$\varphi_i(x) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}} & \text{si } x_{i-1} < x \leq x_i \\ \frac{x - x_i}{x_{i+1} - x_i} & \text{si } x_i < x \leq x_{i+1} \\ 0 & \text{si } x \leq x_{i-1} \quad \text{ou} \quad x_{i+1} < x \end{cases}$$

En utilisant 2 points à l'intérieur de l'intervalle[0, 3]

# **Chapitre 7**

## **DERIVATION NUMERIQUE**

## Chapitre 7 DERIVATION NUMERIQUE

### 7.1 Introduction

Parmi les méthodes de résolution des équations différentielle est la « méthode des différences finies » est la plus facile, elle repose sur l'application du développement en séries de Taylor pour calculer numériquement les valeurs approchées des dérivées.

### 7.2 Définition de la méthode des différence finies

#### 7.2.1 Introduction

On considère la fonction  $y = f(x)$  définie sur  $[a, b]$ . Soit l'ensemble des points  $\{x_0; x_1; x_2; \dots; x_n\}$  équidistants  $x_i = x_0 + ih$ . On note  $y_i = f(x_i)$  la valeur de  $f$  au point  $x_i$  une différence finie est une expression de la forme  $df = f(x') - f(x)$

### 7.3 Opérateurs de différences finies

On note  $y_i = f(x_i)$

#### 7.3.1 Opérateur différence avant

$$\begin{aligned} \Delta f(x) = f(x+h) - f(x) &\Rightarrow \Delta f(x_i) = f(x_{i+1}) - f(x_i) \\ \Delta y_i = y_{i+1} - y_i \end{aligned} \quad (\text{VII. 1})$$

#### 7.3.2 Opérateur différence arrière

$$\begin{aligned} \nabla f(x) = f(x) - f(x-h) &\Rightarrow \nabla f(x_i) = f(x_i) - f(x_{i-1}) \\ \nabla y_i = y_i - y_{i-1} \end{aligned} \quad (\text{VII. 2})$$

#### 7.3.3 Opérateur différence centrée

$$\begin{aligned} \delta f(x) = f(x+h) - f(x-h) &\Rightarrow \delta f(x_i) = f(x_{i+1}) - f(x_{i-1}) \\ \delta y_i = y_{i+1} - y_{i-1} \end{aligned} \quad (\text{VII. 3})$$

Ces opérateurs sont linéaires. Les différences ci-dessus sont la **première différence**. On peut définir des différences d'ordre supérieur. Par exemple la seconde différence est définie par :

$$\begin{aligned} \Delta^2 f(x) = \Delta(\Delta f(x)) &= \Delta f(x+h) - \Delta f(x) \\ \Delta^2 f(x_i) = \Delta(\Delta f(x_i)) &= \Delta f(x_{i+1}) - \Delta f(x_i) \\ \Delta^2 y_i = \Delta y_{i+1} - \Delta y_i \end{aligned} \quad (\text{VII. 4})$$

De manière générale, la  $n^{\text{ième}}$  différence :

$$\begin{aligned} \Delta^n f(x_i) = \Delta^{n-1}(\Delta f(x_i)) &= \Delta^{n-1} f(x_{i+1}) - \Delta^{n-1} f(x_i) \\ \Delta^n y_i = \Delta^{n-1} y_{i+1} - \Delta^{n-1} y_i \end{aligned} \quad (\text{VII. 5})$$

## 7.4 Relation avec la dérivée d'une fonction

### 7.4.1 Théorème 1

Si  $f: \mathbb{R} \rightarrow \mathbb{R}$  est une fonction deux fois (02) continument dérivable (de classe  $C^2$ ) si  $\begin{cases} x_0 \in \mathbb{R} \\ h_0 > 0 \end{cases}$  alors  $\exists c /$

$$\begin{aligned} \left| f'(x_0) - \frac{\nabla_h f(x_0)}{h} \right| &< c \times h ; \forall h \leq h_0 \\ \left| f'(x_0) - \frac{\Delta_h f(x_0)}{h} \right| &< c \times h ; \forall h \leq h_0 \end{aligned} \tag{VII. 6}$$

### 7.4.2 Théorème 2

Si  $f: \mathbb{R} \rightarrow \mathbb{R}$  est une fonction trois fois (03) continument dérivable (de classe  $C^3$ ) si  $\begin{cases} x_0 \in \mathbb{R} \\ h_0 > 0 \end{cases}$  alors  $\exists c /$

$$\left| f'(x_0) - \frac{\delta_h f(x_0)}{h} \right| < c \times h^2 ; \forall h \leq h_0 \tag{VII. 7}$$

### 7.4.3 Définition

On dit que  $\frac{\nabla_h f(x_0)}{h}$  ;  $\frac{\Delta_h f(x_0)}{h}$  sont des formules de différences finies rétrogrades (resp. progressive) pour l'approximation de  $f'(x_0)$ . Les différences 6 et 7 sont appelées erreur de troncature d'ordre  $h$ .  $\frac{\delta_h f(x_0)}{h}$  est la formule de différence centrée où l'erreur de troncature est d'ordre 2.

Si  $f$  est une fonction de classe  $C^{m+1}$  ou de classe  $C^{m+2}$  alors les quantités (VII. 8) sont des approximations de la nième dérivée  $f^{(n)}(x_0)$  au point  $x_0$  quand  $h \rightarrow 0$

$$\frac{\nabla_h^n f(x_0)}{h^n} ; \frac{\Delta_h^n f(x_0)}{h^n} ; \frac{\delta_h^n f(x_0)}{(2h)^n} \tag{VII. 8}$$

On pourrait alors approximer la première dérivée par l'une des formules suivantes :

$$\begin{aligned} f'(x) &= \lim_{h \rightarrow 0} \underbrace{\frac{f(x+h) - f(x)}{h}}_{\text{avant}} = \lim_{h \rightarrow 0} \underbrace{\frac{f(x) - f(x-h)}{h}}_{\text{arrière}} = \lim_{h \rightarrow 0} \underbrace{\frac{f(x+h) - f(x-h)}{2h}}_{\text{centré}} \\ f'(x) &= \begin{cases} \frac{f(x+h) - f(x)}{h} = \frac{\Delta f(x)}{h} \\ \frac{f(x) - f(x-h)}{h} = \frac{\nabla f(x)}{h} \\ \frac{f(x+h) - f(x-h)}{2h} = \frac{\delta f(x)}{2h} \end{cases} \end{aligned} \tag{VII. 9}$$

En général on peut approcher la nième dérivée par :

$$f^{(n)}(x) \cong \frac{\nabla_h^n f(x_0)}{h^n} ; \frac{\Delta_h^n f(x_0)}{h^n} \text{ ou } \frac{\delta_h^n f(x_0)}{(2h)^n} \tag{VII. 10}$$

On peut aussi utiliser une combinaison de ces operateurs tel que :

$$f''(x) \cong \frac{\nabla^2 f(x)}{h^2} \cong \frac{\Delta^2 f(x)}{h^2} \cong \frac{(\Delta \nabla) f(x)}{h^2} \quad (\text{VII. 11})$$

### Cas d'une fonction à plusieurs variables

Soit  $f(x, y)$  une fonction continue dérivable par rapport à  $x$  et  $y$ , on a :

$$\frac{\partial f(x, y)}{\partial x} = \lim_{h \rightarrow 0} \frac{f(x+h, y) - f(x, y)}{h}$$

$$\frac{\partial f(x, y)}{\partial y} = \lim_{k \rightarrow 0} \frac{f(x, y+k) - f(x, y)}{k}$$

Ces dérivées sont approximées par des différences finies

$$\frac{\partial f(x, y)}{\partial x} \approx \frac{f(x+h, y) - f(x, y)}{h}$$

$$\frac{\partial f(x, y)}{\partial y} \approx \frac{f(x, y+k) - f(x, y)}{k}$$

## 7.5 Programme de la méthode de « Dérivation Numérique »

```
!Programme pour calculer numeriquement la derivee d'une fonction f(x)
!-----aux differents pts a<xi<b d'un intervalle [a,b]
implicit none
integer :: i,n
integer,parameter :: nmax=1000
double precision :: a,b,h
real :: f
double precision,dimension(0:nmax)::x,fprime
write(*,*) 'Donner l''intervalle [a,b] et le nbr de pts n < nmax'
read(*,*) a,b,n
h=(b-a)/n
x(0)=a
do i=1,n-1
!Discretisation de l'interval [a,b]
x(i)=x(0)+i*h
!calcul numerique de la derivee en utilisant
!-----l'operateur difference finie avant
fprime(i)=(f(x(i+1))-f(x(i)))/h
write(*,*) "x(", i, ")=", x(i), " et f'('x", i, ")=", fprime(i)
enddo
end

function f(x)
double precision x
f=exp(x)
return
end
```

Figure 7.1 : Programme de la méthode de « Dérivation Numérique »

### Exercice 1

- 1- Définir les opérateurs de différence finie avant  $\Delta$  et arrière  $\nabla$  et centré  $\delta$  respectivement.
- 2- Calculer  $\Delta f(x)$ ,  $\nabla f(x)$  et  $\delta f(x)$  au point  $x = 1$  avec un pas  $h = 0,1$  pour  $f(x) = x^2$
- 3- Calculer l'action générale de l'opérateur  $\Delta\delta$  sur une fonction  $f(x)$ .
- 4- Ecrire un programme fortran qui permet d'écrire  $\Delta f(x)$  en tout point  $x_i$  d'un intervalle  $[a, b]$  :  $(a < x_i < b)$

### Exercice 2

- 1- Calculer l'action de l'opérateur  $D = \nabla\Delta^2$  sur une fonction  $f(x)$  où  $\Delta$  et  $\nabla$  sont les opérateurs différence finie avant et arrière respectivement.
- 2- Donner l'expression approchée de la dérivée troisième  $f'''(x)$  en fonction de l'opérateur  $D$ .

- 3- En déduire la valeur approchée de la dérivée troisième de  $f(x) = e^x$  au point  $x = 1$  si le pas  $h = 0,1$ , comparez cette valeur au résultat exacte.
- 4- Montrer que  $\Delta^n y_i = \nabla^n y_{i+n}$

**Exercice 3**

- 1- Calculer l'action de l'opérateur  $D = \nabla\Delta$  sur une fonction  $f(x)$  où  $\Delta$  et  $\nabla$  sont les opérateurs différence finie avant et arrière respectivement.
- 2- Donner l'expression approchée de la dérivée seconde  $f''(x)$  en fonction de l'opérateur  $D$ .
- 3- En déduire la valeur approchée de la dérivée seconde de  $f(x) = x^2$  au point  $x = 1$  si le pas  $h = 0,1$

**Exercice 4**

- 1- Définir les opérateurs de différence finie avant  $\Delta$  et arrière  $\nabla$  respectivement.
- 2- Montrer que  $\Delta\nabla = \nabla\Delta$



## **Chapitre 8**

# **RESOLUTION NUMERIQUE DES EQUATIONS DIFFERENTIELLES**

## Chapitre 8 RESOLUTION NUMERIQUE DES EQUATIONS DIFFERENTIELLES

### 8.1 Définition

En mathématique, les équations différentielles sont des équations qui lient une fonction ou plus à leurs dérivées. En application physique ou chimique, ces fonctions représentent des grandeurs physique (telle que la masse, la distance, l'énergie, etc.) et leurs dérivées représentent le changement ou l'évolution de ces grandeurs par rapport à un paramètre précis (temps, température, etc.). Dans ce chapitre, on s'intéresse à la résolution des équations différentielles par la méthode des différences finies.

#### 8.1.1 Equation différentielle ordinaire

Plusieurs problèmes en physique nécessitent la résolution d'équation différentielle dont la solution analytique exacte est difficile ou impossible à calculer. On fait alors recours aux méthodes numériques pour trouver les solutions à cette équation :

$$F(y(x); y'(x); y''(x); \dots; y^{(n)}(x)) = 0 \quad (\text{VIII.1})$$

On a en générale 2 approches pour la résolution :

- a) La première consiste à chercher une fonction approchée de la solution et calculer sa valeur aux points  $\{x_0, x_1, x_2, x_3, \dots, x_n\}$ .
- b) La deuxième consiste à calculer les valeurs numériques de la solution  $y(x)$  aux points  $\{x_0, x_1, x_2, x_3, \dots, x_n\}$  espacés par  $h \Rightarrow x_{i+1} = x_i + h$

Equation du premier (1<sup>er</sup>) ordre – « **Problème de Cauchy** » :

$$\begin{cases} \frac{dy}{dx} = g(x, y) \\ y(x_0) = y_0 \end{cases} \quad (\text{VIII.2})$$

#### 8.1.2 Existence et Unicité de la solution

Soit le problème de Cauchy (VIII. 13) tel que :

- a)  $g(x, y)$  est une fonction à valeur réelle définie continue sur l'intervalle  $[x_0, b]$  et  $-\infty < y < +\infty$
- b)  $\exists$  une constante  $k \forall x \in [x_0, b]$  et  $\forall y_1$  et  $y_2 \in \mathbb{R} : \|g(x, y_1) - g(x, y_2)\| \leq k \|y_1 - y_2\|$

Alors  $\forall y_0$  le problème (VIII. 13) admet une solution unique  $y(x) \forall x \in [x_0, b]$  (condition de Lipchitz). Il y'a plusieurs méthodes pour la résolution numérique de ce problème on peut citer :

### 8.2 Méthode de Série de Taylor

#### 8.2.1 Formulation

Basée sur le développement de Taylor :

$$y(x) = \sum_0^{+\infty} a_n (x - x_0)^n \quad (\text{VIII.3})$$

$$a_n = \frac{y(x_0)^{(n)}}{n!}$$

$$y_{i+1} = y_i + h \times y'(x_i) + h^2 \times \frac{y''(x_i)}{2!} + \dots + h^n \times \frac{y^{(n)}(x_i)}{n!} \quad (\text{VIII.4})$$

### 8.3 Méthode de Piccard

#### 8.3.1 Formulation

$$y'(x) = g(x, y) \Rightarrow \int_{y_0}^y dy = \int_{x_0}^x g(x', y) dx' \quad (\text{VIII.5})$$

$$y(x) = y(x_0) + \int_{x_0}^x g(x, y(x')) dx'$$

$$y^1(x) = y(x_0) + \int_{x_0}^x g(x', y(x_0)) dx'$$

$$y^2(x) = y^1(x) + \int_{x_0}^x g(x', y^1(x)) dx'$$

### 8.4 Application de la « Méthode des Différences Finies »

#### 8.4.1 L'algorithme

Les étapes principales de la résolution numérique d'équations différentielles passent par le biais de différents opérateurs de différences finies :

- Discrétisation du domaine  $\Rightarrow$  choix d'un ensemble de point  $\{x_i ; i = 1 \dots n\}$
- Ecrire l'équation différentielle en chaque point  $x_i$  en remplaçant la dérivée par la valeur approchée calculée par  $\Delta, \nabla$  et  $\delta$  ou leur combinaison.
- Déduire le schéma récursif (itératif) qui permet de calculer les valeurs  $y_i$  de la solution  $y(x)$  en chaque point  $x_i$

Soit une équation différentielle munie d'une certaine condition initiale  $y(x_0) = y_0$  tel que :  $g : R \times R \rightarrow R$  une application continue, une solution de cette équation est par définition une application différentiable :

$y(x) : [a, b] \rightarrow R$  vérifiant :

$$\begin{aligned} y'(x) &= g(x, y) \\ y(x_0) &= y_0 \end{aligned} \quad (\text{VIII.6})$$

Ce problème admet comme on l'a déjà souligné plus haut une solution unique si la condition de Lipchitz de rapport  $k$  est vérifiée :

$$\forall x \in [a, b] \text{ et } \forall y_1 \text{ et } y_2 \in \mathbb{R} : \|g(x, y_1) - g(x, y_2)\| \leq k \|y_1 - y_2\|$$

Alors le problème admet une solution unique donnée par :

$$y(x) = y(x_0) + \int_{x_0}^x g(x, y(x')) dx' \quad (\text{VIII.7})$$

## 8.4.2 Méthode d'Euler

### 8.4.2.1 Définition

La méthode consiste à trouver une solution approchée de l'équation sur un intervalle  $[a, b]$  vérifiant les conditions initiales  $y(x_0) = y_0$  en utilisant la méthode des différences finies  $\frac{dy}{dx} \approx \frac{\Delta y}{\Delta x}$

On découpe l'intervalle  $[a, b]$  à l'aide des points  $\{x_0 = a; x_1; x_2; x_3; \dots; x_n = b\}$  en  $n$  sous-intervalles  $\Delta x = h = \frac{x_n - x_0}{n}$  en utilisant l'opérateur différence avant on note

$$\Delta f(x_i) = f(x_{i+1}) - f(x_i) \Rightarrow \Delta y_i = y_{i+1} - y_i \quad (\text{VIII.8})$$

En chaque point  $x_i$  on remplace la dérivée  $y'$  par le rapport des différences finies  $\frac{dy}{dx} = g(x, y)$  par

$$\frac{\Delta y}{\Delta x} = g(x, y) \Rightarrow \frac{\Delta y}{\Delta x} |_{(x, y)} = g(x, y) \quad (\text{VIII.9})$$

En utilisant les opérateurs différence finie  $\frac{dy}{dx} \approx \frac{\Delta y}{h}$

$$\frac{\Delta y_i}{h} = g(x_i, y_i) \quad (\text{VIII.10})$$

## 8.5 Equation Différentielle du 1<sup>er</sup> Ordre

La forme générale d'une équation différentielle d'ordre 1 : (**Méthode Euler**)

$$y'(x) + a(x)y(x) = b(x) \quad (\text{VIII.11})$$

Ecrire sous forme normale :

$$y'(x) = g(x, y) ; \quad \begin{matrix} y(0) = y_0 \\ x \in [a, b] \end{matrix} \text{condition initiale} \quad (\text{VIII.12})$$

On procède comme suit :

$$\begin{aligned} y'_i &= g(x_i, y_i) \\ \Rightarrow \frac{\Delta y_i}{h} &= g(x_i, y_i) \\ \Rightarrow \frac{y_{i+1} - y_i}{h} &= g(x_i, y_i) \end{aligned} \quad (\text{VIII.13})$$

### Schéma récursif

$$y_{i+1} = h \times g(x_i, y_i) + y_i \quad (\text{VIII.14})$$

Ce schéma permet de connaître la valeur approchée de la solution  $y_i$  en chaque point  $x_i$ . Connaissant  $y_0$  on calcule  $y_1$  puis  $y_2$  et ainsi de suite... On calcul ainsi les valeurs approchées aux points en réunissant les points sur le plan  $xOy$  par des segments de droites, nous obtenons une ligne brisée qui est la représentation approchée de la courbe de la solution. Cette ligne s'appelle « **Ligne brisée d'Euler** ». Soit  $y_h(x)$  la solution approchée, on a alors  $\lim_{h \rightarrow 0} |y_h(x) - y_{\text{exa}}(x)| = 0$  où  $y_{\text{exa}}(x)$  est la solution exacte.

### Exercice

Résoudre numériquement par la méthode des différences finies l'équation différentielle suivante avec les conditions citées

$$y'(x) = y(x) + x \quad \text{avec} \quad \begin{cases} x \in [0,1] \\ y(0) = y_0 = 1 \\ h = 0,1 \end{cases}$$

Ecrire un programme Fortran pour la résolution d'équation de type générale :  $y' = g(x,y)$  avec la condition initiale  $y_0$  et  $x \in [a,b]$

### 8.5.1 Programme correspondant à une Equation Différentielle du 1er Ordre – « Méthode d'Euler »

```
!Programme pour résoudre une equation différentielle du 1er ordre de la forme:
!                                     y'(x)=f(x,y) par la methode D'Euler
implicit none
integer :: i,n
integer,parameter :: nmax=1000
double precision :: a,b,h,f
double precision,dimension(0:nmax)::x,y
write(*,*) 'Donner l''intervale [a,b] et le nbr de pts de discretisation n < nmax'
read(*,*) a,b,n
h=(b-a)/n
x(0)=a
y(0)=0.0
!Ouverture d'un fichier pour stocker les résultats
open(2 , file='result.txt', status='New')
write(2,*) x(0),y(0)
!Calcul des x(i) et y(i) par le schéma récursif
do i=1,n
x(i)=x(0)+i*h
y(i)=h*f(x(i-1),y(i-1))+y(i-1)
write(2,*) x(i) , y(i)
end do
close(2)
end
!La fonction
function f(x,y)
doubleprecision::x,y,f
f=y+x
return
end
```

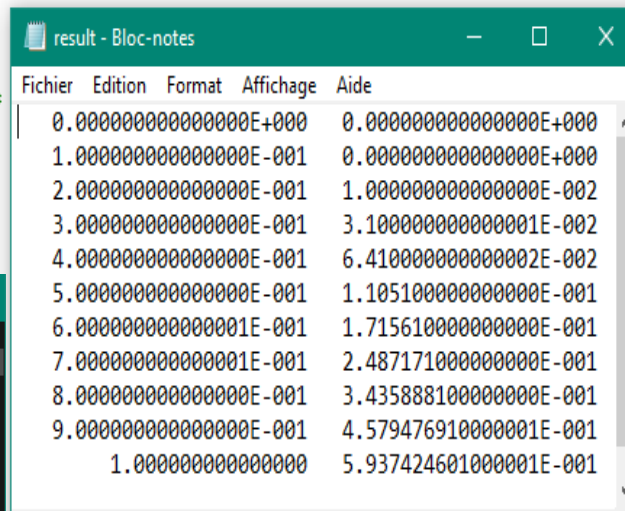


Figure 8.1 : Programme de la « méthode d'Euler » pour une équation différentielle du 1<sup>er</sup> ordre

## 8.6 Equation Différentielle du 2<sup>nd</sup> Ordre

La forme générale d'une équation différentielle d'ordre 2 est :

$$y''(x) + a(x)y'(x) + b(x)y(x) = c(x) \tag{VIII. 15}$$

### 8.6.1 Problème avec Conditions Initiales

$$x \in [a, b] ; y(a) = y_0 ; y'(a) = y'_0$$

$$y''(x_i) = y''_i = \frac{\Delta^2 y_i}{h^2} \tag{VIII. 12}$$

- Première possibilité (double utilisation de l'opérateur avant – ou bien opérateur arrière) :

$$\begin{aligned} \Delta^2 y_i &= \Delta(y'_i) = \Delta(y_{i+1} - y_i) = \Delta y_{i+1} - \Delta y_i = (y_{i+2} - y_{i+1}) - (y_{i+1} - y_i) \\ &\Rightarrow \Delta^2 y_i = y_{i+2} - 2y_{i+1} + y_i \end{aligned} \tag{VIII. 13}$$

- Deuxième possibilité (opérateur mixte) :

$$\Delta(\nabla y_i) = \Delta(y_i - y_{i-1}) = \Delta y_i - \Delta y_{i-1} = (y_{i+1} - y_i) - (y_i - y_{i-1})$$

$$\Rightarrow \Delta(\nabla y_i) = y_{i+1} - 2y_i + y_{i-1} \quad (\text{VIII. 14})$$

8.6.1.1 Recherche du schéma récursif

$$y''(x) + a(x)y'(x) + b(x)y(x) = c(x)$$

On cherche les  $y_i$  en tous les points :  $x_i = [x_0 = a ; x_1 ; x_2 ; x_3 ; \dots ; x_n = b]$

- Première possibilité :

$$\frac{\Delta^2 y_i}{h^2} + a(x_i) \frac{\Delta y_i}{h} + b(x_i)y(x_i) = c(x_i)$$

$$\frac{y_{i+2} - 2y_{i+1} + y_i}{h^2} + a(x_i) \frac{y_{i+1} - y_i}{h} + b(x_i)y(x_i) = c(x_i)$$

**Schéma Récursif**

$$y_{i+2} = h^2 c_i + [2 - ha_i]y_{i+1} + [ha_i - h^2 b_i - 1]y_i \quad (\text{VIII. 15})$$

**Remarque**

On remarque que le premier terme qu'on peut calculer à partir du schéma récursif est  $y_2$ . On aura alors besoin de la valeur de  $y_1$  pour effectuer ce calcul. On peut déterminer  $y_1$  à partir des conditions initiales :  $y(a) = y_0 ; y'(a) = y'_0$

$$y'_0 = \frac{\Delta y_0}{h} = \frac{y_1 - y_0}{h} \Rightarrow y_1 = hy'_0 + y_0 \quad (\text{VIII. 16})$$

- Deuxième possibilité :

$$y''_i = \frac{\Delta(\nabla y_i)}{h^2} \Rightarrow \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + a_i \frac{y_{i+1} - y_i}{h} + b_i y(x_i) = c_i$$

$$\Rightarrow [1 + ha_i]y_{i+1} = h^2 c_i + [2 + ha_i - h^2 b_i]y_i - y_{i-1}$$

**Schéma Récursif**

$$y_{i+1} = \frac{[h^2 c_i + [2 + ha_i - h^2 b_i]y_i - y_{i-1}]}{1 + ha_i} \quad (\text{VIII. 17})$$

**Remarque**

Le premier terme à calculer à partir du schéma récursif est  $y_1$  mais pour le calculer il faudra déterminer  $y_{-1}$  à partir des conditions initiales :  $y(a) = y_0 ; y'(a) = y'_0$

$$y'_0 = \frac{\nabla y_0}{h} = \frac{y_0 - y_{-1}}{h} \Rightarrow y_{-1} = -hy'_0 + y_0 \quad (\text{VIII. 18})$$

**Exemple**

Déterminer à l'aide de la méthode des différences finies le schéma récursif qui permet de résoudre numériquement l'équation différentielle suivante

$$y''(x) - 4y'(x) + y(x) = x$$

Si  $\begin{cases} h = 0,1 \\ y(0) = 0 \text{ et } y'(0) = 1 \end{cases}$  Calculer la valeur approchée de la solution aux points  $x = 0,1$  et  $x = 0,2$

**Solution****- Schéma récursif**

On part d'un point « i » quelconque de l'intervalle et on introduit les opérateurs de différence finie, on obtient l'équation suivante :

$$\Delta^2 y_i'' - 4\Delta y_i' + y_i = x_i$$

A partir de là, on détermine facilement le schéma récursif suivant :

$$y_{i+2} = h^2 x_i + (4h + 2)y_{i+1} - (h^2 + 4h + 1)y_i$$

**- Calcul de y(0,1) et y(0,2)**

On a les conditions initiales suivantes :  $\begin{cases} h = 0,1 \\ y(0) = 0 \text{ et } y'(0) = 1 \end{cases}$

D'après le schéma récursif, quand  $i = 0$

$$y_2 = h^2 x_0 + (4h + 2)y_1 - (h^2 + 4h + 1)y_0$$

Avec  $y_2 = y(0,2)$  ;  $y_1 = y(0,1)$  ;  $y_0 = y(0)$  et  $x_0 = 0$

**- Déterminer  $y_1 = y(0,1)$** 

On sait que  $y'(0) = y'_0 = 1$ . On utilise alors l'opérateur avant pour déterminer le  $y_1$

$$y'_0 = \frac{\Delta y_0}{h} = \frac{1}{h}(y_1 - y_0) \Rightarrow y_1 = y(0,1) = h y'_0 + y_0 = 0,1 * 1 + 0 \Rightarrow \mathbf{y_1 = y(0,1) = 0,1}$$

**- Déterminer  $y_2 = y(0,2)$** 

$$\begin{aligned} y_2 &= h^2 x_0 + (4h + 2)y_1 - (h^2 + 4h + 1)y_0 \\ &= (0,1)^2 \times 0 + (4 \times 0,1 + 2) \times 0,1 - (0,1^2 + 4 \times 0,1 + 1) \times 0 \end{aligned}$$

$$\mathbf{y_2 = y(0,2) = 0,24}$$

**Exercice**

- Quelles sont les étapes principales de la résolution d'une équation différentielle ordinaire par la méthode des différences finies ?
- On considère le problème aux valeurs initiales suivant :  $y'' + xy = 0$  avec  $y(0) = 1$  et  $y'(0) = 0$ . En utilisant les opérateurs de différence finie arrière et avant, déterminer le schéma récursif adéquat qui permet de calculer les valeurs numériques approchées de la solution  $y(x)$
- En déduire  $y(0,2)$  et  $y(0,4)$

8.6.2 Programme correspondant à une Equation du 2<sup>nd</sup> Ordre avec Conditions

Initiales

```
!Programme pour resoudre une equation differentielle du 2nd ordre de la forme:
!----- y''(x)+f(x)y'(x)+g(x)y(x)=d(x) par la methode des differences finies

implicit none
integer :: i,n
integer,parameter :: nmax=1000
double precision :: a,b,h,yp0
real :: f,g,d
double precision,dimension(0:nmax)::x,y
write(*,*) 'Donner l''intervale [a,b] et le nbr n < nmax'

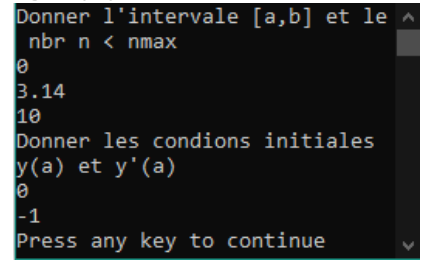
read(*,*) a,b,n

write(*,*) 'Donner les condions initiales y(a) et y''(a)'
read(*,*) y(0),yp0
h=(b-a)/n
x(0)=a
y(1)=h*yp0+y(0)
x(1)=x(0)+h
open(2, file='result2.txt', status='New')
write(2,*) x(0) , y(0)
write(2,*) x(1) , y(1)
do i=1,n
x(i)=x(0)+i*h
!Schéma récursif
y(i+2)=h**2*d(x(i))+(2.-h*f(x(i)))*y(i+1)+(h*f(x(i))-h**2*g(x(i))-1)*y(i)
write(2,*) x(i) , y(i)
enddo
close(2)
end

! Sous programmes associés aux differentes fonctions f(x),g(x) et d(x)
! Ici l'exemple de l'oscillateur harmonique est utilisé
function f(x)
double precision x
f=0.d0
return
endfunction

function g(x)
double precision x
g=1.d0
return
endfunction

function d(x)
double precision x
d=0.d0
return
endfunction
```



result2 - Bloc-notes	
Fichier	Aide
0.000000000000000E+000	0.000000000000000E+000
3.140000000000000E-001	-3.140000000000000E-001
3.140000000000000E-001	-3.140000000000000E-001
6.280000000000000E-001	0.000000000000000E+000
9.420000000000000E-001	3.449591440000000E-001
1.256000000000000	6.899182880000000E-001
1.570000000000000	1.000865840238176
1.884000000000000	1.243790208952704
2.198000000000000	1.388033209283109
2.512000000000000	1.409643470171614
2.826000000000000	1.294399208757641
3.140000000000000	1.040169739758628

Figure 8.2 : Programme de la méthode des « Différences Finies » pour une équation différentielle du 2<sup>nd</sup> ordre

Remarque

Dans ce programme, nous avons utilisé un exemple de l'oscillateur harmonique en appliquant le schéma récursif (VIII. 30).

8.6.3 Problème aux limites

$$y''(x) + a(x)y'(x) + b(x)y(x) = c(x) \tag{VIII. 19}$$

Avec  $x \in [a, b] \begin{cases} y(a) = \delta_1 \\ y'(a) = \delta_n \end{cases}$



$$y_i'' = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}$$

$$\left. \begin{aligned} y_i' &= \frac{\Delta y_i}{h} = \frac{y_{i+1} - y_i}{h} \\ y_i' &= \frac{\nabla y_i}{h} = \frac{y_i - y_{i-1}}{h} \end{aligned} \right\} \Rightarrow y_i' = \frac{y_{i+1} - y_{i-1}}{2h}$$

En remplaçant ces deux formules dans (VII. 15)

$$\Rightarrow \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + a_i \left[ \frac{y_{i+1} - y_{i-1}}{2h} \right] + b_i y(x_i) = c_i \tag{VIII. 20}$$

En rassemblant les termes :

$$q_i y_{i-1} + p_i y_i + r_i y_{i+1} = d_i \tag{VIII. 21}$$

Avec  $\begin{cases} q_i = 2 - ha_i \\ p_i = 2h^2 b_i - 4 \\ r_i = 2 + ha_i \\ d_i = 2h^2 c_i \end{cases}$  et  $y_0 = y(x_0) = y(a) = \delta_1$

$$\begin{aligned} q_1 y_0 + p_1 y_1 + r_1 y_2 &= d_1 \Rightarrow p_1 y_1 + r_1 y_2 = d_1 - q_1 \delta_1 \\ q_2 y_1 + p_2 y_2 + r_2 y_3 &= d_2 \\ q_3 y_2 + p_3 y_3 + r_3 y_4 &= d_3 \\ &\vdots \\ &\vdots \\ q_{n-1} y_{n-2} + p_{n-1} y_{n-1} + r_{n-1} y_n &= d_{n-1} \\ q_{n-1} y_{n-2} + p_{n-1} y_{n-1} &= d_{n-1} - r_{n-1} \delta_n \end{aligned} \tag{VIII. 22}$$

On écrit ce système (VII.26) sous forme matricielle :  $[A][Y] = [F]$

$$\begin{pmatrix} p_1 & r_1 & 0 & 0 & \dots \\ q_2 & p_2 & r_2 & 0 & \dots \\ 0 & q_3 & p_3 & r_3 & \dots \\ \vdots & \vdots & \vdots & \vdots & \dots \\ \vdots & \vdots & \vdots & \vdots & \dots \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-1} \end{pmatrix} = \begin{pmatrix} F_1 \\ F_2 \\ F_3 \\ \vdots \\ F_{n-1} \end{pmatrix} \tag{VIII. 23}$$

Avec  $F_1 = d_1 - q_1 \delta_1$  ;  $F_2 = d_2$  ;  $F_3 = d_3$  ; ... ;  $F_{n-1} = d_{n-1} - r_{n-1} \delta_n$

On recherche la solution "y" en résolvant le système linéaire par les différentes méthodes

(VIII. 24)

**8.6.3.1 Théorème**

On suppose que  $f(x, y, y')$  est continue sur la région  $\mathbb{R} = \{(x, y, y') : a \leq x \leq b ; -\infty \leq y \leq +\infty ; -\infty \leq y' \leq +\infty\}$  et  $f_y$  et  $f_{y'}$  sont continues si  $\exists M > 0$  tel que  $f_y ; f_{y'} > 0 \forall x, y, y' \in \mathbb{R}$  et  $|f(x, y, y')| < M \forall x, y, y' \in \mathbb{R}$  alors  $y'' = f(x, y, y')$  avec  $y(a) = \gamma_1 ; y(b) = \gamma_2$ . Dans ce cas,  $f(x, y, y')$  possède une solution unique  $y(x)$  pour  $a \leq x \leq b$

**8.6.3.2 Méthode du Tir (problème avec conditions aux limites)**

Cette méthode est constituée de trois (03) étapes :

- a. Transformation du problème avec conditions aux limites en 2 problèmes aux valeurs initiales.

- b. Détermination des solutions de ces 2 problèmes.
- c. Recombinaison de ces 2 solutions pour obtenir la solution du problème initiale.

**Exemple**

Soit le problème aux limites :  $y''(x) + a(x)y'(x) + b(x)y(x) = c(x)$  avec  $x \in [a, b]$  et  $\begin{cases} y(a) = \alpha \\ y(b) = \beta \end{cases}$

- Soit  $u(x)$  une solution du problème

$$u''(x) + a(x)u'(x) + b(x)u(x) = c(x) \quad \text{tel que} \quad \begin{cases} u(a) = \alpha \\ u'(a) = 0 \end{cases}$$

- Soit  $v(x)$  aussi une solution du problème

$$v''(x) + a(x)v'(x) + b(x)v(x) = 0 \quad \text{tel que} \quad \begin{cases} v(a) = 0 \\ v'(a) = 1 \end{cases}$$

- La combinaison des deux solutions

$$y(x) = c_1u(x) + c_2v(x) \tag{VIII. 25}$$

Avec

$$\begin{aligned} \alpha &= c_1u(a) + c_2v(a) & \beta &= c_1u(b) + c_2v(b) \\ c_1 &= 1 & c_2 &= \frac{\beta - u(b)}{v(b)} \end{aligned} \tag{VIII. 26}$$

On résout les systèmes d'équations :

$$\begin{cases} u'(x) = w(x) \\ w(a) = u'(a) = 0 \\ w'(x) = -a(x)w(x) - b(x)u(x) + c(x) \end{cases} \tag{VIII. 27}$$

$$\begin{cases} v'(x) = z(x) \\ z(a) = v'(a) = 1 \\ z'(x) = -a(x)z(x) - b(x)v(x) \end{cases} \tag{VIII. 28}$$

**Exemple**

$$y''(x) = y(x) - x \quad \text{avec} \quad y(0) = y'(1) = 0$$

**8.7 Equations aux Dérivées Partielles**

Soit  $f(x, y)$  une fonction continue dérivable par rapport à  $x$  et  $y$ , on a :

$$\begin{aligned} \frac{\partial f(x, y)}{\partial x} &= \lim_{h \rightarrow 0} \frac{f(x + h, y) - f(x, y)}{h} \\ \frac{\partial f(x, y)}{\partial y} &= \lim_{k \rightarrow 0} \frac{f(x, y + k) - f(x, y)}{k} \end{aligned} \tag{VIII. 29}$$

Ces dérivées sont approximées par des différences finies

$$\begin{aligned} \frac{\partial f(x, y)}{\partial x} &\approx \frac{f(x + h, y) - f(x, y)}{h} \\ \frac{\partial f(x, y)}{\partial y} &\approx \frac{f(x, y + k) - f(x, y)}{k} \end{aligned} \tag{VIII. 30}$$

**8.7.1 Résolution d'Equations aux Dérivées Partielles**

La forme générale d'une équation aux dérivées partielles est :

$$f\left(u(x_1, x_2, \dots, x_n), \frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2}, \frac{\partial^2 u}{\partial x_1^2}, \frac{\partial^2 u}{\partial x_2^2}, \dots\right) = 0 \tag{VIII. 31}$$

Une équation d'ordre 2 est de la forme générale suivante :

$$A(x, y) \frac{\partial^2 U(x, y)}{\partial x^2} + B(x, y) \frac{\partial^2 U(x, y)}{\partial x \partial y} + C(x, y) \frac{\partial^2 U(x, y)}{\partial y^2} + D(x, y) \frac{\partial U(x, y)}{\partial x} + E(x, y) \frac{\partial U(x, y)}{\partial y} + F(x, y)U(x, y) = G(x, y) \tag{VIII. 32}$$

On a 3 types selon le signe de  $\Delta = B^2 - 4AC$

$\Delta > 0 \Rightarrow$  équation elliptique

$\Delta = 0 \Rightarrow$  équation parabolique (VIII. 33)

$\Delta < 0 \Rightarrow$  équation hyperbolique

**Exemples**

**Exemple 1**

Résoudre dans le domaine  $D = [a, b] \times [c, d]$  avec  $x \in [a, b]$  et  $y \in [c, d]$  l'équation différentielle suivante :  $\frac{\partial f}{\partial x}(x, y) + \frac{\partial f}{\partial y}(x, y) = xy$

- On commence par discrétiser le domaine  $D = [a, b] \times [c, d]$  c.à.d. les intervalles des deux variables. On obtient :

$$\begin{cases} x \in [a, b] \Rightarrow x_i = x_0 + ih / x_0 = a \\ y \in [c, d] \Rightarrow y_j = y_0 + jk / y_0 = c \end{cases}$$

- On écrit l'équation en un nœud quelconque  $(x_i, y_j)$

$$\frac{\partial f}{\partial x}(x_i, y_j) + \frac{\partial f}{\partial y}(x_i, y_j) = x_i y_j$$

- On remplace les dérivées par leurs expressions approchées en utilisant les opérateurs différences finies :

$$\begin{aligned} \frac{\partial f(x_i, y_j)}{\partial x} &\approx \frac{\Delta_x f(x_i, y_j)}{h} = \frac{f(x_i + h, y_j) - f(x_i, y_j)}{h} = \frac{f(x_{i+1}, y_j) - f(x_i, y_j)}{h} \\ \frac{\partial f(x_i, y_j)}{\partial y} &\approx \frac{\Delta_y f(x_i, y_j)}{k} = \frac{f(x_i, y_j + k) - f(x_i, y_j)}{k} = \frac{f(x_i, y_{j+1}) - f(x_i, y_j)}{k} \end{aligned}$$

Avec la notation simplificatrice  $f(x_i, y_j) = f_{i,j}$  on aura :

$$\frac{\partial f(x_i, y_j)}{\partial x} \approx \frac{f_{i+1,j} - f_{i,j}}{h} \quad \text{et} \quad \frac{\partial f(x_i, y_j)}{\partial y} \approx \frac{f_{i,j+1} - f_{i,j}}{k}$$

L'équation différentielle s'écrit alors comme suit :

$$\frac{f_{i+1,j} - f_{i,j}}{h} + \frac{f_{i,j+1} - f_{i,j}}{k} = x_i y_j \quad (*)$$

- L'équation de **Laplace** est une équation elliptique  $\Rightarrow \Delta(\mathbf{u}(x, y, z)) = 0$

**Exemple 2**

- L'équation de la **chaleur** est une équation parabolique :

$$\frac{\partial u(x, y)}{\partial y} = a^2 \frac{\partial^2 u(x_i, y_j)}{\partial x^2}$$

On dérive par rapport à  $x$  et  $y$  tel que  $x \in [a, b]$  et  $y \in [c, d]$

- Première dérivée :

$$\begin{cases} \frac{\partial u}{\partial x} = \frac{u(x+h, y) - u(x, y)}{h} \\ \frac{\partial u}{\partial y} = \frac{u(x, y+k) - u(x, y)}{k} \end{cases} \quad \text{(VIII. 34)}$$

- Seconde dérivée :

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial}{\partial x} \left( \frac{\partial u}{\partial x} \right) = \frac{\frac{\partial u}{\partial x}(x+h, y) - \frac{\partial u}{\partial x}(x, y)}{h}$$

Quand on utilise la dérivée arrière, on trouve :

$$\frac{\partial^2 u}{\partial x^2} = \frac{u(x+h, y) - 2(u(x, y)) + u(x-h, y)}{h^2} \quad \text{(VIII. 35)}$$

Si on discrétise l'intervalle des  $x \in [a, b]$  en  $x_i : x_0 = a, x_1 = x_0 + h, \dots, x_i = x_0 + ih, \dots, x_n = b$

Si on discrétise l'intervalle des  $y \in [c, d]$  en  $y_j : y_0 = c, y_1 = y_0 + k, \dots, y_j = y_0 + jk, \dots, y_m = d$

Donc on a :

$$\frac{\partial^2 u}{\partial x^2}(x_i, y_j) = \frac{u(x_i+h, y_j) - 2(u(x_i, y_j)) + u(x_i-h, y_j)}{h^2}$$

$$\frac{\partial^2 u}{\partial x^2}(x_i, y_j) = \frac{u(x_{i+1}, y_j) - 2(u(x_i, y_j)) + u(x_{i-1}, y_j)}{h^2} \quad \text{(VIII. 36)}$$

On a 
$$\begin{aligned} x_i &= x_0 + ih & h &= \frac{b-a}{n} \\ y_j &= y_0 + jk & k &= \frac{c-d}{m} \end{aligned}$$

$$\frac{\partial^2 u_{ij}}{\partial x^2} = \frac{u_{i+1j} - 2u_{ij} + u_{i-1j}}{h^2} \quad \text{(VIII. 37)}$$

Même chose pour  $y$

$$\frac{\partial^2 u_{ij}}{\partial y^2} = \frac{u_{ij+1} - 2u_{ij} + u_{ij-1}}{k^2} \quad \text{(VIII. 38)}$$

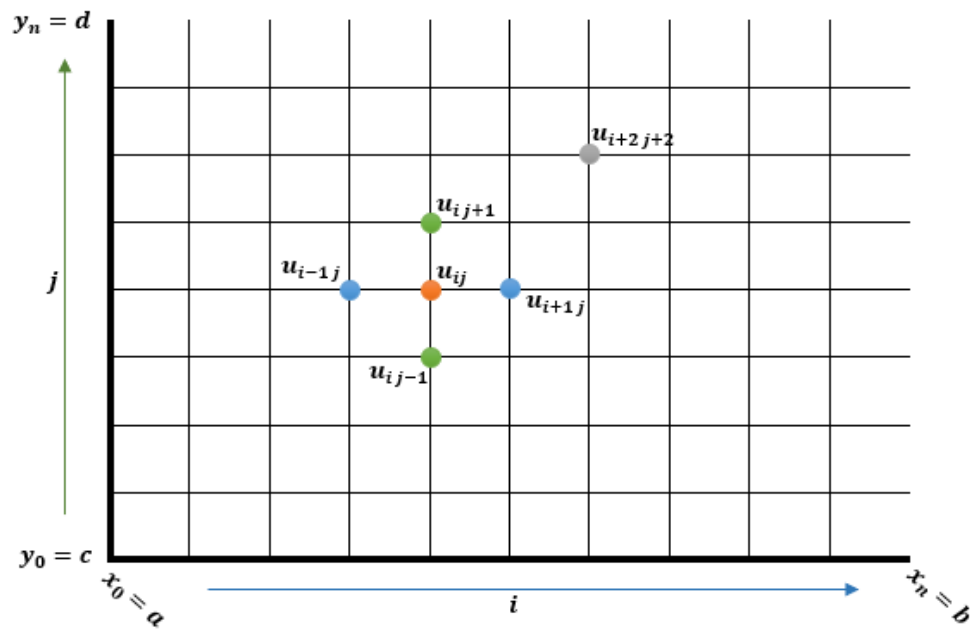


Figure 8.3: Schéma démonstratif des points de discrétisation dans le plan (xOy)

**Exemple**

Equation de la propagation de chaleur : On chauffe l'extrémité d'une barre métallique



$$\frac{\partial u(x, t)}{\partial t} = a^2 \frac{\partial^2 u(x_i, t_j)}{\partial x^2} \tag{VIII. 39}$$

Avec  $a^2$  : coefficient de propagation correspondant au métal utilisé

Après application des opérateurs des différences finies, on obtient :

$$\frac{u_{ij+1} - u_{ij}}{k} = a^2 \frac{u_{i+1j} - 2u_{ij} + u_{i-1j}}{h^2}$$

$$u(x, 0) = \varphi(x)$$

Les conditions aux limites sont :  $u(L, t) = \Psi_1(x)$

$$u(0, t) = \Psi_2(x)$$

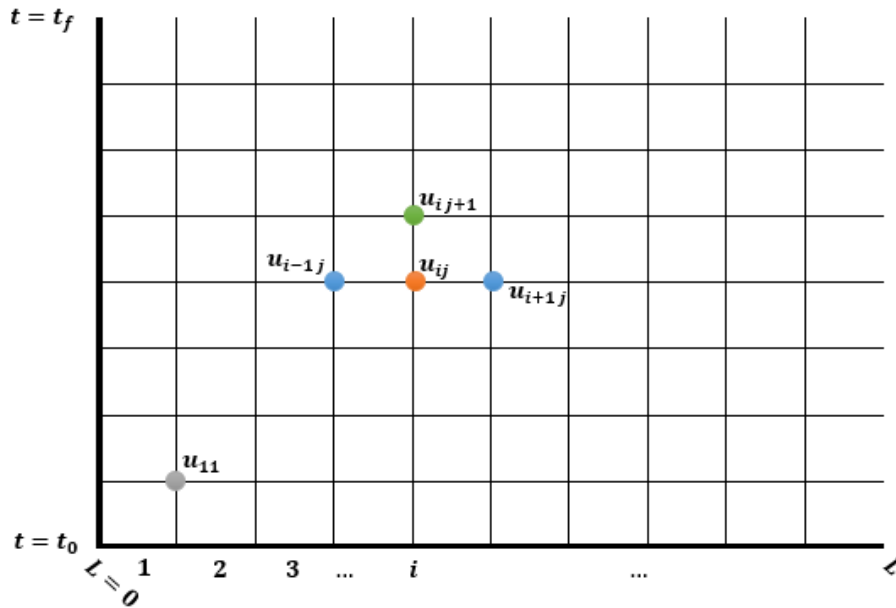


Figure 8.4 : Schéma démonstratif des points de discrétisation pour l'exemple de l'équation de chaleur

Donc le terme qu'on peut déterminer à partir des précédents est

$$u_{ij+1} = \frac{ka^2}{h^2} [u_{i+1j} + u_{i-1j}] + \left[ \frac{1}{k} - \frac{2a^2}{h^2} \right] u_{ij} \tag{VIII. 40}$$

N.B.

La connaissance des valeurs de  $u(x, t)$  aux points  $(u_{i+1j}), (u_{i-1j}), (u_{ij})$  permet de calculer la valeur de  $u(x, t)$  au point  $(i + 1j)$

**Exemple 2**

Soit l'équation aux dérivées partielles  $\frac{\partial U(x,y)}{\partial x} + \frac{\partial^2 U(x,y)}{\partial x \partial y} = xy$

1. En utilisant les opérateurs de différence finie montrer qu'on a :

$$\frac{\partial^2 U(x, y)}{\partial x \partial y} \approx \frac{U(x + h, y + k) - U(x, y + k) - U(x + h, y) + U(x, y)}{hk}$$

2. Discrétiser, trouver le schéma récursif puis résoudre numériquement l'équation (calculer la valeur numérique approchée de  $U(x, y)$  aux différents nœuds) si  $0 \leq x \leq 1$  et  $0 \leq y \leq 1$  et  $h = k = 0,5$  avec les conditions aux limites suivantes :  $U(0, y) = U(1, y) = U(x, 0) = 1$  Faire un schéma.

**Solution**

1. On utilise l'opérateur avant et on obtient :

$$\begin{aligned} \frac{\partial^2 U(x, y)}{\partial x \partial y} &\approx \frac{\Delta_x \Delta_y U(x, y)}{hk} = \frac{\Delta_x}{hk} (\Delta_y U(x, y)) = \frac{\Delta_x}{hk} (U(x, y + k) - U(x, y)) \\ &= \frac{\Delta_x U(x, y + k) - \Delta_x U(x, y)}{hk} \end{aligned}$$

On aura alors le résultat voulu  $\frac{\partial^2 U(x,y)}{\partial x \partial y} \approx \frac{U(x+h,y+k)-U(x,y+k)-U(x+h,y)+U(x,y)}{hk}$

2. a. Discrétisation du domaine  $D = [0, 1] \times [0, 1]$  (voir figure 7.3)

$$\begin{aligned} x_i = x_0 + ih &\Rightarrow x_0 = 0; x_1 = 0,5; x_2 = 1 \\ y_j = y_0 + jk &\Rightarrow y_0 = 0; y_1 = 0,5; y_2 = 1 \end{aligned}$$

2. b. Résolution numérique de l'équation : On doit définir le schéma récursif à partir de l'équation différentielle exprimée en un nœud quelconque  $(x_i, y_j)$

$$\frac{\partial U(x_i, y_j)}{\partial x} + \frac{\partial^2 U(x_i, y_j)}{\partial x \partial y} = x_i y_j$$

En utilisant les opérateurs de différence finie, on obtient l'expression approchées des dérivées :

$$\begin{aligned} \frac{\partial U(x_i, y_j)}{\partial x} &\approx \frac{\Delta_x U(x_i, y_j)}{h} = \frac{U(x_i + h, y_j) - U(x_i, y_j)}{h} = \frac{U_{i+1j} - U_{ij}}{h} \\ \frac{\partial^2 U(x_i, y_j)}{\partial x \partial y} &\approx \frac{U_{i+1j+1} - U_{ij+1} - U_{i+1j} + U_{ij}}{hk} \quad (\text{d'après question 1}) \end{aligned}$$

L'équation discrétisée devient :

$$\frac{U_{i+1j} - U_{ij}}{h} + \frac{U_{i+1j+1} - U_{ij+1} - U_{i+1j} + U_{ij}}{hk} = x_i y_j (**)$$

Les nœuds mis en jeu sont ceux schématisé dans la figure suivante.

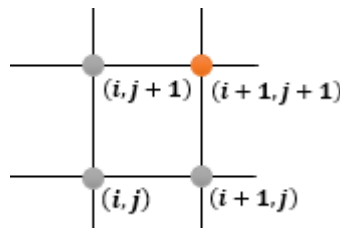


Figure 8.5 : Disposition des différents nœuds qui figurent dans l'expression (\*\*)

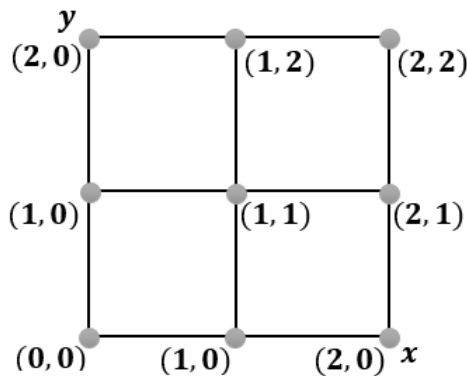


Figure 8.6 : Discrétisation du domaine  $D = [0, 1] \times [0, 1]$

Maintenant on doit trouver quel nœud il faut exprimer en fonction des autres. D'après la figure 7.2 et les conditions aux limites du problèmes, il est clair qu'on exprime soit  $U_{i+1j+1}$  ou  $U_{ij+1}$  en fonction des autres :  $U_{i+1j+1} = hk(x_i y_j) + (k - 1)(U_{ij} - U_{i+1j}) + U_{ij+1}$

Les conditions aux limites  $U(x, 0) = 0 \forall x \in [0, 1]$  donne les valeurs de  $U(x, y)$  aux nœuds de la première rangée des  $x : (i, j) = \{(0,0), (1,0), (2,0)\}$

Il faut se rappeler que  $U(x_i, y_j) = U_{ij}$  la valeur de  $U(x, y)$  au nœud  $(i, j)$

$$U(x_0, y_0) = U_{00} = U(x = 0, y = 0) = 1 \quad \text{La valeur de } U(x, y) \text{ au nœud } (0, 0)$$

$$U(x_1, y_0) = U_{10} = U(x = 0,5, y = 0) = 1 \quad \text{La valeur de } U(x, y) \text{ au nœud } (1, 0)$$

$$U(x_2, y_0) = U_{20} = U(x = 1, y = 0) = 1 \quad \text{La valeur de } U(x, y) \text{ au nœud } (2, 0)$$

**C.à.d.  $U_{i0} = 1$  pour  $i = 0, 1, 2$**

De même les conditions aux limites  $U(0, y) = 0 \forall y \in [0, 1]$  donne les valeurs de  $U(x, y)$  aux nœuds de la première rangée des  $y : (i, j) = \{(0, 0), (0, 1), (0, 2)\}$

$$U(x_0, y_0) = U_{00} = U(x = 0, y = 0) = 1 \quad \text{La valeur de } U(x, y) \text{ au nœud } (0, 0)$$

$$U(x_0, y_1) = U_{01} = U(x = 0, y = 0,5) = 1 \quad \text{La valeur de } U(x, y) \text{ au nœud } (0, 1)$$

$$U(x_0, y_2) = U_{02} = U(x = 0, y = 1) = 1 \quad \text{La valeur de } U(x, y) \text{ au nœud } (0, 2)$$

**C.à.d.  $U_{0j} = 1$  pour  $j = 0, 1, 2$**

De même les conditions aux limites  $U(1, y) = 0 \forall y \in [0, 1]$  donne les valeurs de  $U(x, y)$  aux nœuds de la rangée verticale des  $y : (i, j) = \{(2, 0), (2, 1), (2, 2)\}$

$$U(x_2, y_0) = U_{20} = U(x = 1, y = 0) = 1 \quad \text{La valeur de } U(x, y) \text{ au nœud } (2, 0)$$

$$U(x_2, y_1) = U_{21} = U(x = 1, y = 0,5) = 1 \quad \text{La valeur de } U(x, y) \text{ au nœud } (2, 1)$$

$$U(x_2, y_2) = U_{22} = U(x = 1, y = 1) = 1 \quad \text{La valeur de } U(x, y) \text{ au nœud } (2, 2)$$

**C.à.d.  $U_{2j} = 1$  pour  $j = 0, 1, 2$**

Il reste maintenant les valeurs de  $U$  aux nœuds  $(1, 1)$  et  $(2, 1)$  on les calcule à partir du schéma récursif

$$U_{i+1j+1} = hk(x_i y_j) + (k - 1)(U_{ij} - U_{i+1j}) + U_{ij+1}$$

On donne  $i = 0$  et  $j = 0$  on aura  $U_{11}$  puis  $i = 1$  et  $j = 0$  on aura  $U_{21}$

$$U_{11} = hk(x_0 y_0) + (k - 1)(U_{00} - U_{10}) + U_{01} = 1 \Rightarrow U_{11} = 1$$

$$U_{21} = hk(x_1 y_0) + (k - 1)(U_{10} - U_{20}) + U_{11} = 1 \Rightarrow U_{21} = 1$$

### Exemple 3

Soit l'équation suivante  $\frac{\partial^2 U(x, t)}{\partial t^2} = \frac{\partial^2 U(x, t)}{\partial x^2}$

- 1- Si  $0 \leq x \leq 1$  et  $0 \leq t \leq 0,5$  discrétiser le domaine  $(x, t)$  en calculant les valeurs des nœuds  $(x_i, t_j)$  tel que  $x_i = x_0 + ih$  et  $t_j = t_0 + jk$  avec  $h = 0,2$  et  $k = 0,1$
- 2- En utilisant les opérateurs de différence finie arrière et avant montrer que pour une fonction  $f(x, y)$  on a

$$\frac{\partial^2 f(x, y)}{\partial x^2} \approx \frac{f(x + h, y) - 2f(x, y) + f(x - h, y)}{h^2}$$

- 3- Déterminer le schéma récursif qui relie la valeur inconnue de  $U(x, t)$  en un nœud en fonction de ses valeurs connues aux autres nœuds. Faire un schéma.
- 4- Résoudre numériquement l'équation (calculer la valeur numérique approchée de  $U(x, t)$  aux différents nœuds) avec les conditions aux limites suivantes :

$$U(0, t) = U(1, t) = 0 ; U(x, 0) = \frac{1}{2} \sin(\pi x) ; U(x_i, t_1) = U(x_i, t_{-1}) \Leftrightarrow U_{i1} = U_{i-1}$$

### Solution



1- Discrétisation du domaine  $D = [0; 1] \times [0; 0,5]$

$$x_i = x_0 + ih \Rightarrow \{x_0 = 0; x_1 = 0,2; x_2 = 0,4; x_3 = 0,6; x_4 = 0,8; x_5 = 1\}$$

$$t_j = t_0 + jk \Rightarrow \{t_0 = 0; t_1 = 0,1; t_2 = 0,2; t_3 = 0,3; t_4 = 0,4; t_5 = 0,5\}$$

2- Voir le cours pour les détails pour la démonstration de la formule suivante :

$$\frac{\partial^2 f(x, y)}{\partial x^2} \approx \frac{\Delta_x \nabla_x f(x, y)}{h^2} = \frac{f(x + h, y) - 2f(x, y) + f(x - h, y)}{h^2}$$

3- Schéma récursif : On doit utiliser la formule de la question précédente

$$\frac{\partial^2 U(x_i, y_j)}{\partial x^2} \approx \frac{\Delta_x \nabla_x U(x_i, y_j)}{h^2} = \frac{U(x_i + h, y_j) - 2U(x_i, y_j) + U(x_i - h, y_j)}{h^2} = \frac{U_{i+1j} - 2U_{ij} + U_{i-1j}}{h^2}$$

$$\frac{\partial^2 U(x_i, y_j)}{\partial t^2} \approx \frac{\Delta_t \nabla_t U(x_i, y_j)}{k^2} = \frac{U(x_i, y_j + k) - 2U(x_i, y_j) + U(x_i, y_j - k)}{k^2} = \frac{U_{ij+1} - 2U_{ij} + U_{ij-1}}{k^2}$$

L'équation différentielle s'écrit alors comme suit :

$$\frac{U_{ij+1} - 2U_{ij} + U_{ij-1}}{k^2} = \frac{U_{i+1j} - 2U_{ij} + U_{i-1j}}{h^2} (*)$$

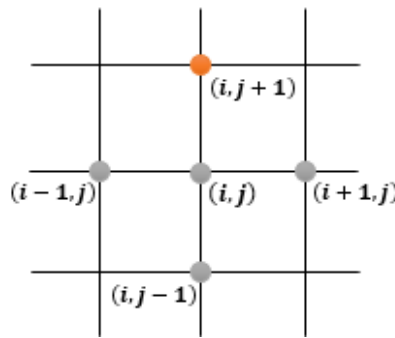


Figure 8.7 : Disposition des différents nœuds qui figurent dans l'expression (\*)

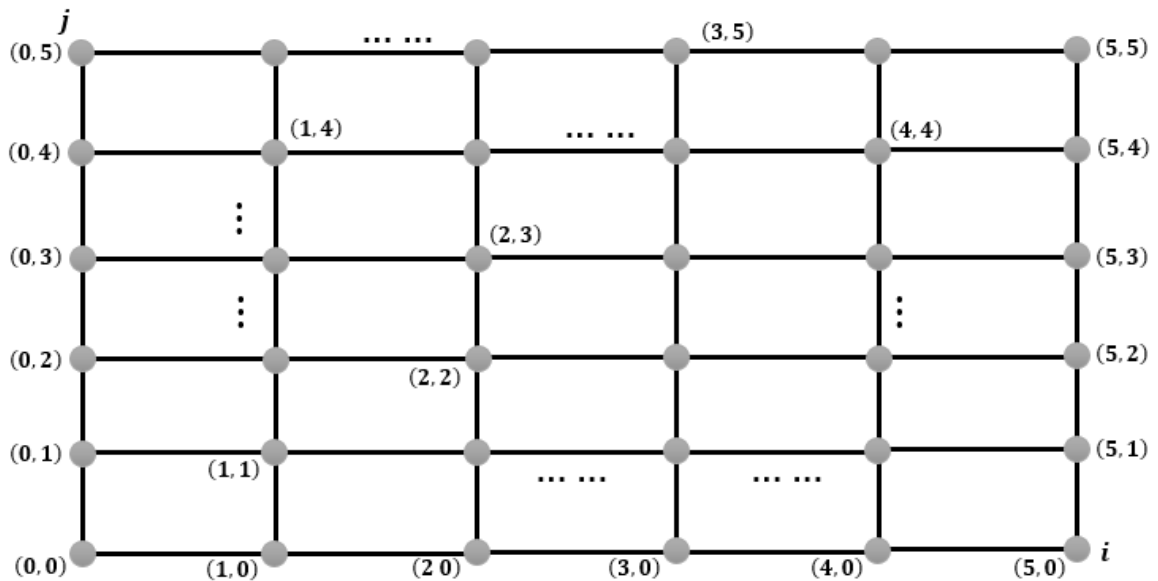


Figure 8.8 : Discrétisation du domaine  $D = [0; 1] \times [0; 0,5]$

Maintenant, exactement comme l'exemple précédent, on doit trouver quel nœud il faut exprimer en fonction des autres. D'après la figure 7.4 et les conditions aux limites du problème, il est clair qu'on exprime soit  $U_{ij+1}$  en fonction des autres.

$$U_{ij+1} = \frac{k^2}{h^2}(U_{i+1j} + U_{i-1j}) + 2\left(1 - \frac{k^2}{h^2}\right)U_{ij} - U_{ij-1}$$

Ce schéma récursif nous permet de calculer les valeurs de  $U(x, t)$  aux différents nœuds  $(i, j)$

Les conditions aux limites  $U(x, 0) = 0 \forall x \in [0, 1]$  donne les valeurs de  $U(x, y)$  aux nœuds de la première rangée des  $x : (i, j) = \{(0, 0), (1, 0), (2, 0), (3, 0), (4, 0), (5, 0)\}$

**Il faut se rappeler que  $U(x_i, t_j) = U_{ij}$  la valeur de  $U(x, t)$  au nœud  $(i, j)$**

$$U(x, 0) = \frac{1}{2}\sin(\pi x) \Rightarrow U(x_i, t_0) = U_{i0} = \frac{1}{2}\sin(\pi x_i)$$

$U(x_0, t_0) = U_{00} = U(x = 0, t = 0) = 0,5\sin(\pi x_0) = 0$	<i>La valeur de <math>U(x, y)</math> au nœud <math>(0, 0)</math></i>
$U(x_1, t_0) = U_{10} = U(x = 0,2, t = 0) = 0,5\sin(\pi x_1) = 0,5877$	<i>La valeur de <math>U(x, y)</math> au nœud <math>(1, 0)</math></i>
$U(x_2, t_0) = U_{20} = U(x = 0,4, t = 0) = 0,5\sin(\pi x_2) = 0,9510$	<i>La valeur de <math>U(x, y)</math> au nœud <math>(2, 0)</math></i>
$U(x_3, t_0) = U_{30} = U(x = 0,6, t = 0) = 0,5\sin(\pi x_3) = 0,9510$	<i>La valeur de <math>U(x, y)</math> au nœud <math>(3, 0)</math></i>
$U(x_4, t_0) = U_{40} = U(x = 0,8, t = 0) = 0,5\sin(\pi x_4) = 0,5877$	<i>La valeur de <math>U(x, y)</math> au nœud <math>(4, 0)</math></i>
$U(x_5, t_0) = U_{50} = U(x = 1, t = 0) = 0,5\sin(\pi x_5) = 0$	<i>La valeur de <math>U(x, y)</math> au nœud <math>(5, 0)</math></i>

De même les conditions aux limites  $U(0, t) = 0 \forall t \in [0 ; 0,5]$  donne les valeurs de  $U(x, t)$  aux nœuds de la première rangée des  $t : (i, j) = \{(0; 0), (0, 1), (0, 2), (0, 3), (0, 4), (0, 5)\}$

$$U(0, t_j) = U(x = 0, t_j) = U(x_0, t_j) = U_{0j} = 0 \text{ pour tout } (0, j) \forall j \in [0, 5]$$

La condition  $U(1, t) = 0 \forall t \in [0 ; 0,5]$  donne les valeurs de  $U(x, t)$  aux nœuds de la dernière rangée des  $t : (i, j) = \{(5; 0), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5)\}$

$$U(1, t_j) = U(x = 1, t_j) = U(x_5, t_j) = U_{5j} = 0 \text{ pour tout } (0, j) \forall j \in [0, 5]$$

**La dernière rangée des  $t$  car  $x = 1$  correspond à  $x_5$**

Il reste maintenant les valeurs de  $U$  aux nœuds  $U_{11}, U_{12}, U_{22}, \dots, U_{43}, \dots, U_{55}$ . On les calcule à partir du schéma récursif

$$U_{ij+1} = \frac{k^2}{h^2}(U_{i+1j} + U_{i-1j}) + 2\left(1 - \frac{k^2}{h^2}\right)U_{ij} - U_{ij-1}$$

On donne  $i = 1$  et  $j = 0$  on aura  $U_{11}$

$$U_{11} = \frac{k^2}{h^2}(U_{20} + U_{00}) + 2\left(1 - \frac{k^2}{h^2}\right)U_{10} - U_{1-1}$$

- D'après les conditions aux limites, on a  $U_{20} = U_{10} = U_{00} = 0$
- Pour  $U_{1-1}$  on utilise la condition  $U_{i-1} = U_{i1}$  d'où  $U_{1-1} = U_{11}$

$$2U_{11} = \frac{k^2}{h^2}(U_{20} + U_{00}) + 2\left(1 - \frac{k^2}{h^2}\right)U_{10}$$

Avec  $U_{00} = 0 ; U_{20} = \frac{1}{2}\sin(\pi x_2) = \frac{1}{2}\sin(0,2\pi) ; U_{10} = \frac{1}{2}\sin(\pi x_1) = \frac{1}{2}\sin(0,1\pi)$

De même pour les autres valeurs :

$$U_{21} = \frac{k^2}{h^2}(U_{30} + U_{10}) + 2\left(1 - \frac{k^2}{h^2}\right)U_{20} - U_{2-1}$$

Et on procède ainsi pour calculer  $U_{22}, \dots, U_{43}, \dots, U_{55}$  jusqu'à la fin.

**Exemple 4**

Résoudre numériquement l'équation de Poisson  $\frac{\partial^2 U(x,y)}{\partial y^2} + \frac{\partial^2 U(x,y)}{\partial x^2} = -2x^2 + y^2$  avec les conditions suivantes :

$$\begin{aligned} 0 \leq x \leq 2 & \quad h = k = 0,5 & \quad U(0, y) = U(2, y) = U(x, 0) = U(x, 2) = 0 \\ 0 \leq y \leq 2 & & \quad U(x_i, y_1) = U(x_i, y_{-1}) \text{ c. à d. } U_{i1} = U_{i-1} \end{aligned}$$

Pour cet exemple, le problème est le même que l'exemple précédent sauf que la variable  $t$  est remplacée par  $y$  et le second membre n'est pas nul, il est égal à  $-2x^2 + y^2$ . Ça n'empêche pas de suivre les mêmes étapes de résolution suivies dans l'exemple précédent.

**Exemple 5**

Appliquer ce que vous avez appris dans ce chapitre aux cas physique concrets suivants :

- a. La radioactivité
- b. Mouvement d'un projectile avec et sans résistance de l'air.
- c. Problème Képlérien (2 masses soumise à un potentiel central de la forme  $1/r$ )
- d. Oscillateur harmonique classique à 1 dimension
- e. Oscillateur harmonique quantique à 1 dimension
- f. Propagation de la chaleur le long d'une barre de longueur  $L$
- g. Propagation d'une onde le long d'une corde avec des extrémités fixes.

**Solution**

Il suffit de trouver les équations différentielles qui régissent ou qui gouvernent les phénomènes cités dans la liste et procéder à la résolution numérique de chaque équation en utilisant les méthodes employées dans ce chapitre.

- a. Prenons l'exemple de la radioactivité. L'équation différentielle de la radioactivité décrit la variation du nombre de noyaux radioactif dans le temps c.à.d.

$$\frac{dN(t)}{dt} = -\lambda N(t) \text{ avec } N(t_0) = N_0$$

C'est une équation du premier ordre avec condition initiale.

- b. Propagation de la chaleur le long d'une barre de longueur  $L$

$$\frac{\partial T(x, t)}{\partial t} = \alpha \frac{\partial^2 T(x, t)}{\partial x^2}$$

On remarque bien que c'est une équation différentielle du second ordre.

- c. Si on prend l'exemple de l'oscillateur harmonique classique à 1 D :

$$y''(x) + \omega^2 y(x) = 0$$

Avec les condition initiales  $y(x_0) = y_0$  et  $y'(x_0) = y'_0$

On remarque que c'est une équation du second ordre avec condition initiales.

On continue de la même manière pour les autres phénomènes. Trouvant chaque équation correspondante et appliquant la résolution numérique de la méthode étudiée dans ce chapitre.

**Chapitre 9**  
**METHODES DE MONTE – CARLO**

---

# Chapitre 9 METHODES DE MONTE – CARLO

## 9.1 Définition

Le véritable développement des méthodes de Monte-Carlo s'est produit pendant de la deuxième guerre mondiale lors des recherches sur la fabrication de la bombe atomique, ce sont Von Neumann, Ulam et Métropolies qui ont largement contribué à l'essor de cette méthode.

Les méthodes de Monte Carlo permettent d'estimer des quantités en utilisant la simulation de variables aléatoires comme le calcul d'intégrales, la résolution de systèmes linéaires, résolution d'équations différentielles et des équations aux dérivées partielles. Son application s'est étendue après au domaine de la physique. Actuellement les méthodes de Monte Carlo forment la classe la plus grande et la plus importante de méthodes numériques utilisées pour résoudre des problèmes de physique statistique.

Les méthodes de Monte-Carlo permettent de calculer une valeur numérique en utilisant des procédés aléatoires, c'est-à-dire des techniques probabilistes. C'est pour cela que le nom de ces méthodes fait allusion aux jeux de hasard pratiqués au Casino de Monte-Carlo à Monaco.

## 9.2 Rappel sur les notions de base de probabilité des variables aléatoires

### 9.2.1 Définition

On appelle une variable aléatoire une grandeur  $x$  associée à une expérience aléatoire. Une expérience est dite « aléatoire » lorsqu'elle a plusieurs résultats possibles et qu'on ne peut pas prévoir celui qui sera obtenu. Le résultat est uniquement dû au hasard.

### 9.2.2 Variables aléatoires discrètes

Les valeurs sont discontinues et dénombrables  $x = \{x_1, x_2, x_3, \dots, x_n\}$ . Par exemple, la valeur de la face obtenue après le jet d'un dé.



*Figure 9.1 : image d'un dé*

Avec  $x = \{x_1 = 1, x_2 = 2, x_3 = 3, x_4 = 4, x_5 = 5, x_6 = 6\}$ . A chaque valeur  $x_i$  est attribuée une probabilité  $p(x = x_i) = p_i$

Dans l'exemple du dé on a :

$$p(x = x_1) = p_1 = \frac{1}{6} ; p(x = x_2) = p_2 = \frac{1}{6} ; \dots ; p(x = x_6) = p_6 = \frac{1}{6}$$

Autre exemple est la **loi de Bernoulli** où on a une variable aléatoire  $x$  qui prend 2 valeurs :  $x = \{0 ; 1\}$

Cette loi décrit des situations avec deux possibilités ou réalisations (échec ou succès), (pile ou face – pour une pièce de monnaie).



**Figure 9.2 :** Schématisation du jet d'une pièce de monnaie (pile ou face)

Si  $p_0$  et  $p_1$  sont les probabilités d'avoir les événements  $x = 0$  et  $x = 1$  respectivement, on a alors

$$p_0 + p_1 = 1 \Leftrightarrow p_1 = 1 - p_0 \quad (\text{IX. 1})$$

Par exemple quand on jette une pièce de monnaie on a  $p_0 = p_1 = \frac{1}{2}$

### 9.2.2.1 L'espérance

L'espérance de la variable aléatoire  $x$  qu'on note  $E[x]$  ou  $\bar{x}$  est :

$$E[x] = \sum_{i=1}^n p_i x_i \quad (\text{IX. 2})$$

Dans le cas du dé par exemple

$$E[x] = \sum_{i=1}^6 p_i x_i = p_1 x_1 + p_2 x_2 + p_3 x_3 + p_4 x_4 + p_5 x_5 + p_6 x_6$$

$$E[x] = \frac{1}{6} \cdot 1 + \frac{1}{6} \cdot 2 + \frac{1}{6} \cdot 3 + \frac{1}{6} \cdot 4 + \frac{1}{6} \cdot 5 + \frac{1}{6} \cdot 6 = \frac{21}{6}$$

#### 9.2.2.1.1 Définition

L'espérance sert donc à prévoir la valeur moyenne obtenue pour la variable que l'on mesure si l'expérience est renouvelée un très grand nombre de fois.

Si  $x$  est une variable aléatoire alors toute fonction  $g(x)$  est aussi une variable aléatoire et on a

$$E[g(x)] = \sum_{i=1}^n p_i g(x_i) \quad (\text{IX. 3})$$

Moment d'ordre  $k$  :  $g(x) = x^k$

$$E[x^k] = \sum_{i=1}^n p_i x_i^k \quad (\text{IX. 4})$$

Moment centré d'ordre  $k$  :  $g(x) = (x - E(x))^k$

$$E[(x - E(x))^k] = \sum_{i=1}^n p_i (x - E(x))^k \quad (\text{IX. 5})$$

### 9.2.2.2 La variance

Le cas le plus important est le moment central d'ordre 2 qui est la « **Variance  $V(x)$**  ». Elle exprime la moyenne des carrés des écarts à la moyenne.

$$k = 2 \Rightarrow V(x) = E[(x - E(x))^2] \quad (\text{IX. 6})$$

On montre que

$$V(x) = E[(x - E(x))^2] = E[x^2] - E[x]^2 \quad (\text{IX. 7})$$

On comprend alors que la variance est aussi égale à la différence entre la moyenne des carrés des valeurs de la variable et le carré de la moyenne. Elle est d'autant plus grande que les valeurs sont étalées. C'est un indicateur de « **dispersion des valeurs** » et leur distribution autour de la moyenne.

On définit aussi l'écart type  $\sigma = \sqrt{V(x)} = \sqrt{E[x^2] - E[x]^2}$

### 9.2.3 Variables aléatoires continues

Dans ce cas la variable  $x$  prends des valeurs continues, comme exemple on peut citer le temps de désintégration d'un atome radioactif et la température d'une région ou d'une ville pendant une certaine période de l'année. La probabilité d'avoir une valeur de  $x$  comprise entre  $[x ; x + dx]$  est

$$p(x) = f(x) dx \quad (\text{IX. 8})$$

Où  $f(x)$  est la densité de probabilité vérifiant les propriétés suivantes :

- $f(x) \geq 0$
- $f(x)$  est continue sauf peut-être en un nombre fini de point
- $\int_{-\infty}^{+\infty} f(x) dx = 1$

La probabilité d'avoir la valeur  $x$  comprise dans un intervalle  $a < x < b$  est :

$$p(a < x < b) = \int_a^b f(x) dx \quad (\text{IX. 9})$$

#### 9.2.3.1 L'espérance

De même on définit l'espérance

$$\begin{cases} E[x] = \int_{-\infty}^{+\infty} x f(x) dx \\ E[g(x)] = \int_{-\infty}^{+\infty} g(x) f(x) dx \end{cases} \quad (\text{IX. 10})$$



$$E[(x - E(x))^2] = \int_{-\infty}^{+\infty} (x - E(x))^2 f(x) dx \quad (\text{IX. 11})$$

Il existe des exemples types pour la densité de distribution comme :

### 9.2.3.2 La distribution uniforme

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{si } x \in [a, b] \\ 0 & \text{ailleurs} \end{cases} \quad (\text{IX. 12})$$

$$E[x] = \int_{-\infty}^{+\infty} x f(x) dx = \int_a^b x \frac{1}{b-a} dx = \frac{(a+b)}{2}$$

### 9.2.3.3 La loi de Poisson

Une variable suit la distribution de Poisson si  $f(x)$  est de la forme

$$f(x = n) = \frac{\lambda^n}{n!} e^{-\lambda} \quad (\text{IX. 13})$$

Où  $\lambda = E[x] = V(x)$

### 9.2.3.4 La loi normale ou la loi de Laplace-Gauss

Dans ce cas la densité de probabilité est de la forme

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x - \bar{x})^2}{2\sigma^2}\right] \quad (\text{IX. 14})$$

### 9.2.3.5 Cas de plusieurs variables aléatoires

La densité de probabilité est  $f(x_1, x_2, \dots, x_n)$  appelée « **densité de probabilité jointe** ». Si les variables  $x_i = \{x_1, x_2, \dots, x_n\}$  sont indépendantes alors  $f(x_1, x_2, \dots, x_n)$  se factorise et on aura :

$$f(x_1, x_2, \dots, x_n) = f_1(x_1) f_2(x_2) \dots f_n(x_n) \quad (\text{IX. 15})$$

## 9.3 Principe de la méthode Monte – Carlo

L'une des procédures pour calculer une quantité par la méthode de Monte – Carlo est de la mettre tout d'abord sous la forme d'une espérance. A l'issue de cette étape, il reste à calculer cette quantité par une espérance  $E(x)$  de la variable aléatoire  $x$ . Pour ce calcul, il convient de savoir simuler une variable aléatoire selon une certaine loi. On dispose alors d'une suite  $x_i (i = 1, \dots, N)$  de  $N$  réalisations de la variable aléatoire  $x$ . On approxime alors  $E(x)$  par :

$$E(x) \approx \frac{x_1 + x_2 + x_3, \dots + x_N}{N} \quad (\text{IX. 16})$$

Cette approximation est garantie par la loi des grands nombres et le théorème de la limite centrale.

### 9.3.1 La loi des grands nombres

Soit une suite de variables aléatoires indépendantes de même loi de distribution tel que  $E[x] = m$  fini en posant

$$S_N = \frac{x_1 + x_2 + x_3, \dots + x_N}{N} \quad (\text{IX. 17})$$

Alors la **probabilité** que  $\lim_{N \rightarrow +\infty} (E(x) - S_N) = 0$  est égale à 1 :

$$\lim_{N \rightarrow +\infty} E(x) \approx \frac{x_1 + x_2 + x_3, \dots + x_N}{N}$$

### 9.3.2 Théorème de la limite centrale

Soit  $x_1, x_2, \dots$  une suite de variables aléatoires réelles indépendantes et identiquement distribuées suivant la même loi. Supposons que l'espérance  $E[x] = m$  et l'écart – type  $\sigma$  existent et soient finis avec  $\sigma \neq 0$ .

Considérons la somme (IX. 17) :

$$S_N = \frac{x_1 + x_2 + x_3 + \dots + x_N}{N}$$

Alors l'espérance de  $S_n$  est  $m$  et son écart-type vaut  $\frac{\sigma}{\sqrt{N}}$

De plus, quand  $N$  est assez grand, la loi normale est une bonne approximation de la loi de  $S_n$ . Ce théorème garanti que l'erreur commise par l'approximation :

$$E(x) \approx \frac{x_1 + x_2 + x_3 + \dots + x_N}{N} \quad (\text{IX. 18})$$

Peut être rendue aussi petite qu'on veut en diminuant l'écart type  $\frac{\sigma}{\sqrt{N}}$  en augmentant  $N$  ou en réduisant la variance  $V(x) = \sigma^2$  c.à.d.

$$E(x) \approx S_n \Leftrightarrow \begin{cases} \frac{\sigma}{\sqrt{N}} \searrow \searrow \\ N \uparrow \\ V(x) = \sigma^2 \searrow \searrow \end{cases} \quad (\text{IX. 19})$$

### 9.4 Intégration par la Méthode Monte – Carlo

Soit  $I = \int_a^b g(x) dx$  l'intégrale d'une fonction continue sur  $[a, b]$ . Supposons qu'on a une variable aléatoire  $x$  avec une densité de probabilité  $f(x)$  sur  $[a, b]$  tel que  $f(x) = 0$  si  $x \notin [a, b]$ . Donc pour une fonction  $h(x)$  de la variable aléatoire  $x$  :

$$E[h(x)] = \int_a^b h(x) f(x) dx \quad (\text{IX. 20})$$

Maintenant on peut écrire notre intégrale  $I$  en introduisant  $f(x)$  :

$$I = \int_a^b g(x) dx = \int_a^b \frac{g(x)}{f(x)} f(x) dx = \int_a^b h(x) f(x) dx \quad (\text{IX. 21})$$

Où  $h(x) = \frac{g(x)}{f(x)}$

Donc  $I = E[h(x)] = E \left[ \frac{g(x)}{f(x)} \right]$ . Mais d'après la loi des grands nombres, on peut approcher  $E[h(x)]$  par :

$$E[h(x)] \simeq \sum_{i=1}^N \frac{h(x_i)}{N} \text{ avec } N \gg \gg \gg \quad (\text{IX. 22})$$

D'où

$$I = \int_a^b g(x)dx \simeq \sum_{i=1}^N \frac{h(x)}{N} \quad (\text{IX. 23})$$

Dans le cas particulier où la loi de distribution est uniforme :

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{si } x \in [a, b] \\ 0 & \text{ailleurs} \end{cases} \quad \text{et } h(x) = \frac{g(x)}{b-a}$$

$$I = \int_a^b g(x)dx \simeq \frac{1}{b-a} \sum_{i=1}^N \frac{g(x)}{N} \quad (\text{IX. 24})$$

#### 9.4.1 Génération de variables aléatoires

Il existe plusieurs méthodes pour générer (produire) des variables aléatoires en utilisant certains algorithmes. Elles sont dites « **pseudo-aléatoires** » pour pouvoir les différencier des variables aléatoires réelles qui sont des processus naturels comme la radioactivité par exemple. Parmi les algorithmes qui génèrent ces variables, il y a ceux qui sont basés sur la « **Relation Congruence** ».

$$x_{n+1} = ax_n[m] \quad (\text{IX. 25})$$

Où  $a$  et  $m$  sont deux entiers positifs. Cette relation montre que  $x_{n+1}$  est le reste de la division :  $ax_n/m$

##### Exemple

Si  $a = 10$  et  $m = 7$  et  $x_0 = 1$  les 3 premières valeurs de la liste des variables aléatoires produit avec cette relation sont :

$$\begin{aligned} x_1 &= 10x_0[7] = 10[7] = 3 \\ x_2 &= 10x_1[7] = 30[7] = 2 \\ x_3 &= 10x_2[7] = 20[7] = 6 \end{aligned} \quad (\text{IX. 26})$$

... et ainsi de suite

Généralement on génère les variables aléatoires dans un intervalle donné qui est  $[0, 1]$  pour cela on divise  $x_{n+1}$  par  $m$  on aura  $0 < x_{n+1}/m < 1$ . Dans l'exemple précédent, les nouvelles variables deviennent  $x_1 = \frac{3}{7}$  ;  $x_2 = \frac{2}{7}$  ;  $x_3 = \frac{6}{7}$  ; ....

Ainsi toute intégrale sur un intervalle  $[a, b]$  doit être transformée sur un intervalle  $[0; 1]$  par un changement de variable :

$$\int_a^b g(x)dx = (b-a) \int_0^1 g(x')dx' \quad (\text{IX. 27})$$

Après le changement de variable

$$x' = \frac{x-a}{b-a} \Rightarrow dx' = \frac{dx}{b-a} \quad (\text{IX. 28})$$

Chaque générateur d'événement a une période. Par définition, la période c'est la séquence ou la liste des nombres après laquelle le premier nombre est reproduit (se répète) une deuxième fois. Dans l'exemple si dessus :

Si  $a = 10$  et  $m = 7$  et  $x_0 = 1$  les 6 premières valeurs de la liste des variables aléatoires produit avec cette relation sont :

$$\begin{array}{ll}
x_1 = 10x_0[7] = 10[7] = 3 & x_4 = 10x_3[7] = 60[7] = 4 \\
x_2 = 10x_1[7] = 30[7] = 2 & x_5 = 10x_4[7] = 40[7] = 5 \\
x_3 = 10x_2[7] = 20[7] = 6 & x_6 = 10x_5[7] = 50[7] = 1
\end{array}
\quad (\text{IX. 29})$$

Donc on retrouve la valeur  $x_0 = 1$  en calculant  $x_6$  ce qui veut dire que la période est jusqu'à  $x_6$ . Pour cela un générateur de nombre aléatoire possède « **Une Graine Aléatoire** (ou état de graine) ». C'est un numéro (ou un vecteur) utilisé pour initialiser le générateur et avoir une nouvelle séquence ou liste de nombre aléatoire.

## 9.5 Algorithme « Génération de Nombres Aléatoires » dans l'Intervalle

### [0, 1] – « Algorithme de Congruence »

Ecrire un programme Fortran permettant de générer des nombres aléatoires (100 nombres par exemple) dans l'intervalle [0, 1] par l'algorithme de congruence  $x_{n+1} = ax_n[m]$ . Les étapes principales de l'algorithme se résument comme suit :

- Lecture des données ( $a ; x_0$  et  $m$ )
- A partir de  $x_0$ , on calcul la suite des  $x_{n+1}$  par la relation  $x_{n+1} = ax_n/m$
- $x_{n+1} = x_{n+1}/m$  pour avoir  $0 < x_{n+1} < 1$
- Affichage du résultat

#### 9.5.1 Fonction et Subroutine de génération de nombre aléatoire en Fortran

Les langages de programmation contiennent des fonctions et des sous-routines spéciales qui permettent de générer des nombres aléatoires. Par exemple en Fortran, on a la fonction

- La fonction « **random** »

$$\text{rand}(\ ) \quad (\text{IX. 30})$$

Elle produit des nombres aléatoires suivant une distribution uniforme dans l'intervalle [0, 1]

- La subroutine

$$\text{random\_number}(\ ) \quad (\text{IX. 31})$$

##### 9.5.1.1 Syntaxe

- « **random** » : La ligne dans le programme (syntaxe)

$$x = \text{rand}(\ ) \quad (\text{IX. 32})$$

Donne un nombre aléatoire entre 0 et 1 qui est attribué à  $x$ .

- La subroutine :

$$\text{call random\_number}(x) \quad (\text{IX. 33})$$

Donne un nombre aléatoire compris entre 0 et 1 qui est attribué à la variable  $x$ .

Donc pour produire plusieurs nombres aléatoires il faut mettre la fonction **rand()** ou la subroutine **random\_number()** dans une boucle Do.

- Initialisation d'un générateur de nombres aléatoires

$$\text{call random\_seed}(\ ) \quad (\text{IX. 34})$$

Effectue une initialisation du générateur des nombres aléatoires pour avoir une nouvelle séquence de nombres aléatoires.

### 9.5.2 Programme de la méthode « Génération des nombres aléatoires » dans [0 ; 1]

```
!programme de generation de nombres aleatoire
implicit none
real y ,rand
integer :: i, n
write(*,*) "Donnez le nombre de valeurs à generer n"
read(*,*) n

Open(4, file="nbr_aleat.txt", status="new")
do i=1,n
  y=rand(i)
  write(4,*) y
enddo
Close(4)
end
```

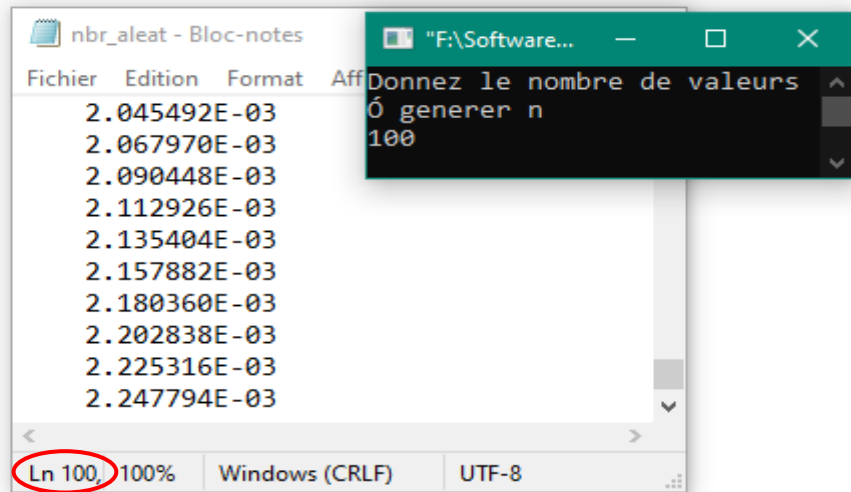


Figure 9.3: Programme de la méthode "Génération des Nombres Aléatoires" avec résultats stockés dans un fichier texte (N.B. le fichier contient bien les 100 nombres demandés)

### 9.5.3 Programme de « l'Algorithme de Congruence »

```
!programme de generation de nombres aleatoire par l'algorithme congruentionnel
!----- x(n+1)=ax(n)[m]
implicit none
real y ,rand
integer :: a,m,n ,i
integer , dimension(0:1000):: x
print*, 'donner la valeur de a, x0 le diviseur m et le nombre de valeur a generer n'
read(*,*) a,x(0),m,n
```

```
do i=1,5
  y=rand(i)
  print*,y
enddo

Open(2, file="Congruence.txt", status="new")
write(2,*) x(0)
do i=0,n
  x(i+1)=mod(a*x(i),m)
  write(2,*) x(i+1)
enddo
Close(2)
end
```

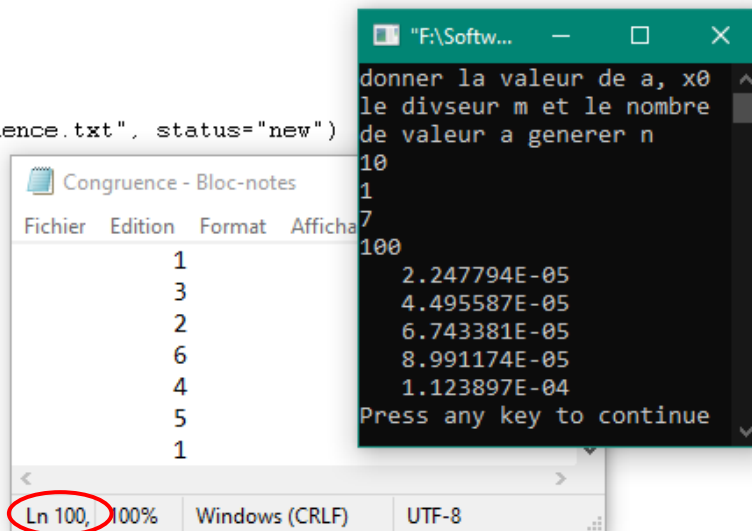


Figure 9.4 : Programme de l'algorithme de Congruence avec affichage de résultat

## 9.6 Algorithme « Calcul d'une Intégrale »

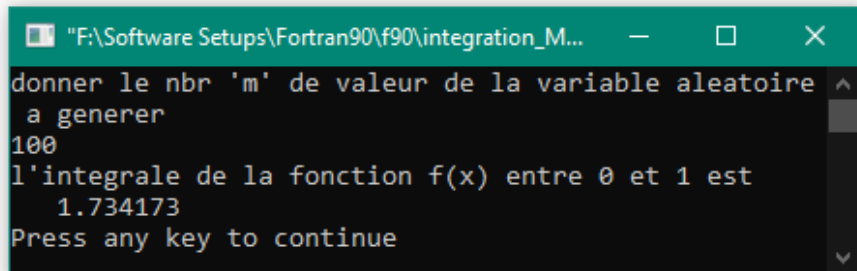
Ecrire un programme Fortran permettant de calculer l'intégrale d'une fonction en utilisant la fonction `rand( )` Ou la subroutine `random_number( )`. Les étapes principales de cet algorithme sont citées ci-dessous :

- La lecture des bornes de l'intégrale et le nombre  $N$  des nombres qu'on veut produire dans l'intervalle  $[0, 1]$
- Générer (produire) des nombres aléatoires par la fonction du Fortran `rand( )`
- Calcul de la somme  $\sum_{i=1}^N \frac{h(x)}{N}$  puis  $I = (b - a) \sum_{i=1}^N \frac{h(x)}{N}$
- Affichage du résultat

### 9.6.1 Programme de la méthode « Monte – Carlo » pour le calcul d'intégrale

```
!programme pour l'integration de fonction entre 0 et 1
implicit none
real y ,rand,f,integrale
integer :: i,m,n
print*,'donner le nbr ''m'' de valeur de la variable aleatoire a generer'
read(*,*) m
n=0
do i=1,m
  y=rand()
  integrale=integrale+f(y)
  n=n+1
enddo
integrale=integrale/n
write(*,*) 'l'integrale de la fonction f(x) entre 0 et 1 est' , integrale
end

function f(x)
f=exp(x)
return
endfunction
```



**Figure 9.5 :** Programme de la méthode "Monte – Carlo" pour le calcul de l'intégrale de la fonction ( $\exp x$ ) entre 0 et 1

#### Exercice 1

- Calculer les 3 premiers nombres aléatoires par l'algorithme  $x_{n+1} = ax_n[m]$  si  $a = 5$  ;  $m = 3$  et  $x_0 = 1$
- Montrer que les valeurs de la nouvelle variable aléatoire définie par  $y_n = \frac{x_n}{m}$  sont comprises entre 0 et 1 ( $0 \leq y_n \leq 1$ )
- Montrer qu'avec la méthode Monte-Carlo on a  $\int_b^a f(x)dx = E \left[ \frac{f(x)}{p(x)} \right]$  où  $x$  est une variable aléatoire générée suivant la densité de probabilité  $p(x)$  et  $E$  désigne l'espérance (la moyenne).
- Avec la loi des grands nombres ( $N \rightarrow \infty$ ) on a  $\left[ \frac{f(x)}{p(x)} \right] \approx \frac{1}{N} \sum_0^N \frac{f(x_i)}{p(x_i)}$

Ecrire dans ce cas un programme Fortran qui calcule  $\int_0^1 f(x)dx$  si la densité  $P(x) = 1$  dans  $[0, 1]$  et nulle ailleurs et en supposant qu'on a généré  $N$  valeurs aléatoire  $y_i$  dans  $[0, 1]$

#### Exercice 2

On considère le générateur de nombres aléatoires basé sur le calcul de congruence  $x_{n+1} = ax_n[m]$

- 1) Calculer les 3 premiers nombres aléatoires  $x_i$  puis  $y_i = \frac{x_i}{m}$  par l'algorithme si
$$a = 7 ; m = 5 \text{ et } x_0 = 3$$
  - 2) Utiliser ces 3 nombres générés pour estimer la valeur de  $\int_0^1 x \, dx$  sachant que la densité de probabilité  $p(x) = 1$  dans  $[0, 1]$
-

## *Références*

- [1] Méthodes numériques : algorithmes, analyse et applications  
Auteurs : Alfio Quaternio, Ricardo Sacco et all  
Edition: Springer 2007
  - [2 ] Méthodes numériques appliquées : avec nombreux problèmes résolus en Fortran IV..  
Auteurs Alain Gourdin, Mohamed Boumahrat  
Editeur : Technique et documentation Paris 1983
  - [3 ] Méthodes numériques et optimisation : théorie et pratique pour l'ingénieur,  
Auteur Jean-Pierre Corriou  
Éd. Tec & doc 2010
  - [4] Méthodes numériques pour l'ingénieur  
Auteurs : Bouchaïb Radi Abdelkhalak el Hami  
Edition Ellipses 2010
  - [5] Analyse numérique : algorithme et étude mathématique ; cours et exercices corrigés  
Auteur : Francis Filbet  
Edition : Dunod 2013
  - [6] Analyse numérique 1  
Auteur : Derradji Salah  
Edition : OPU:Alger 1990
  - [7] Programmer en Fortran 90 : guide complet  
Auteur : Claude Delannoy  
Edition : Eyrolles
  - [8] FORTRAN numerical recipes in Fortran 90  
Auteur William H. Press  
Editeur : Cambridge University Press
-