

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

**UNIVERSITÉ MOHAMED KHIDER, BISKRA**

FACULTÉ des SCIENCES EXACTES et des SCIENCES de la NATURE et de la VIE

**DÉPARTEMENT DE MATHÉMATIQUES**



Mémoire présenté en vue de l'obtention du Diplôme :

**MASTER en Mathématiques**

Option : **Analyse**

Par

**AGTI Nouhed**

Titre :

**Sur la Méthode d'Interpolation Polynomiale**

Membres du Comité d'Examen :

Dr. <b>DAKHIA Hassiba</b>	UMKB	Président
Dr. <b>OUAAR Fatima</b>	UMKB	Encadreur
Dr. <b>BOUZIANE Nadjjet</b>	UMKB	Examineur

Juin 2018

## DÉDICACE

Je dédie cet humble travail

À l'homme de ma vie, mon exemple éternel, mon soutien moral et source de joie et de bonheur, celui qui s'est toujours sacrifié pour me voir réussir, que Dieu te garde à moi, à toi mon père.

À la lumière de mes jours, la source de mes efforts, la flamme de mon coeur, ma vie et mon bonheur, maman que j'adore.

À mes frères et mes sœurs.

À mon cher Mohamed El Amine, que Dieu le bénisse.

À toute ma famille, mes amis et mes collègues de l'université de Biskra.

À tout ce qui ont enseigné moi au long de ma vie scolaire.

## REMERCIEMENTS

Louanges à "Dieu" le tout puissant de m'avoir donné la foi et de m'avoir permis d'en arriver là. Merci "Allah" qui m'a donné la force, la santé, le courage, la patience et la volonté afin d'accomplir ce travail modeste.

J'exprime ma reconnaissance et mes vifs remerciements à mon encadreur Madame "OUAAR Fatima" pour avoir encadré et dirigé avec une grande rigueur scientifique, la qualité de sa formation et ses conseils, le soutien et a confiance qu'elle mes a accordée, j'ai permis de réaliser cette période de recherche dans les meilleures conditions.

Je remercie très sincèrement les membres de Jury d'avoir bien voulu accepter de faire partie de la commission d'examiner, Dr DAKHIA Hassiba le président et Dr BOUZIANE Nadjat l'examineur de ce mémoire.

Nos remerciements vont aussi à toute personne ayant collaboré de près ou de loin à l'amélioration de ce travail et à tous les enseignants du département de mathématiques qui ont contribué à notre formation, et spécialement le chef Dr- Hafayed Mokhtar.

Je tiens à remercier aussi l'université de Mohammed Khider BISKRA de nous donner l'opportunité de vivre une expérience aussi enrichissante tout sur le plan humain que professionnel, tout en gardant la possibilité de poursuivre les études.

Finalement, je tiens à exprimer ma profonde gratitude à ma famille qui m'a toujours soutenu moralement et financièrement, et a tous mes amis et collègues à l'université de Mohamed Khider Biskra.

# Table des matières

Dédicace	i
Remerciements	ii
Table des matières	ii
Liste des figures	v
Introduction	1
<b>1 Généralité</b>	<b>3</b>
1.1 Espace des polynômes . . . . .	3
1.2 Problème général de l'interpolation polynomaile . . . . .	5
1.2.1 Position du problème d'interpolation . . . . .	5
<b>2 Méthodes d'interpolation</b>	<b>10</b>
2.1 Méthodes d'interpolation . . . . .	10
2.1.1 Interpolation de Lagrange (1736 – 1813) . . . . .	10
2.1.2 Interpolation de Newton (1643 – 1727) . . . . .	15
2.1.3 Interpolation d'Hermite ou Osculateur (1822 – 1901) . . . . .	20
2.1.4 Interpolation par les fonctions splines . . . . .	23
2.2 Erreur d'interpolation . . . . .	28
2.2.1 Erreur d'interpolation de Lagrange . . . . .	28

2.2.2	Erreur d'interpolation d'Hermite . . . . .	30
2.2.3	Erreur d'interpolation par les fonctions splines . . . . .	30
<b>3</b>	<b>Applications avec Matlab</b>	<b>32</b>
3.1	Application de la méthode de Lagrange . . . . .	33
3.2	Application de la méthode de Newton . . . . .	35
3.3	Application de la méthode d'Hermite . . . . .	37
3.4	Application de la méthode des fonctions splines . . . . .	40
3.4.1	Par spline linéaire . . . . .	40
3.4.2	Par spline cubique . . . . .	42
	<b>Conclusion</b>	<b>45</b>
	<b>Bibliographie</b>	<b>47</b>
	<b>Annexe : Logiciel <i>Matlab</i></b>	<b>49</b>

# Table des figures

1.1	Problème d'interpolation. . . . .	6
2.1	Polynômes de Lagrange de degré 1 : $L_0(x)$ et $L_1(x)$ . . . . .	12
2.2	Polynôme de Lagrange de degré 2 : $L_0(x)$ , $L_1(x)$ et $L_2(x)$ . . . . .	13
3.1	Interpolation selon Lagrange de $y = \exp(x)$ . . . . .	34
3.2	Interpolation selon Newton de $y = \exp(x)$ . . . . .	36
3.3	Interpolation selon Hermite de $y = \exp(x)$ . . . . .	39
3.4	Interpolation par splines linéaires de $y = \exp(x)$ . . . . .	41
3.5	Interpolation par splines cubique de $y = \exp(x)$ . . . . .	44
3.6	Comparaison entre la solution exacte ( $y = \exp(x)$ ) et leurs solutions appro- chées. . . . .	45

# Introduction

Le développement de l'analyse numérique, et des mathématiques en général, a été et sera toujours nécessaire pour la résolution des problèmes de plus en plus complexes posés par la physique et les sciences de l'ingénieur.

On peut dire que l'analyse numérique est à la frontière entre les mathématiques et l'informatique. Il s'agit en quelque sorte d'un interprète qui permettra de transposer la connaissance mathématique théorique à la pratique d'un ordinateur et de pouvoir ainsi résoudre des problèmes concrets. Les outils de l'analyse numérique sont utilisés dans toutes les disciplines et sont appliqués dans toutes les technologies : bureau d'étude pour la conception des appareils ou machines que nous rencontrons dans la vie courante (four à micro ondes, automobiles, avions, smartphones...), appareils médicaux comme les scanners, contrôle dans les machines de travaux publics, prévision du temps etc ... Une formation mathématique équilibrée devrait comporter une initiation aux techniques du calcul scientifique, quel que soit l'emploi auquel on se destine. Classiques de l'analyse numérique, mises en œuvre pour la résolution de problèmes concrets issus de la physique, la mécanique, la chimie, le traitement de l'image, etc.

Les deux objectifs principaux de l'analyse numérique sont, d'une part, de pouvoir résoudre numériquement des problèmes concrets dont on connaît ou pas la solution analytique et d'autre part, d'analyser le comportement des méthodes utilisées.

On sait bien qu'il est facile d'évaluer un polynôme quand on connaît ses coefficients. En tant qu'outil théorique, l'interpolation résout le problème inverse et montre qu'un

polynôme de degré  $n$  peut aussi bien se représenter par ses  $(n + 1)$  coefficients que par ses valeurs en  $(n + 1)$  points quelconques. Cette observation est la base des algorithmes les plus rapides connus pour la multiplication des polynômes. Cela nous conduit à poser la question suivante :

### **A quoi l'interpolation sert-elle ?**

L'interpolation consiste à trouver une fonction passant exactement par les points à notre disposition, elle apparaît aussi comme sous problème dans de nombreux problèmes plus appliqués. Par exemple, pour intégrer une expression compliquée, il suffit de l'évaluer en  $(n+1)$  points, d'approximer l'expression par un polynôme de degré  $n$  (interpolations) et d'intégrer ce polynôme.

Pour répondre à cette problématique, on divise mon travail à trois chapitres :

**Chapitre 1 :** nous présentons quelques définitions générales des polynômes, l'interpolation et la matrice de Vandermonde.

**Chapitre 2 :** nous visons plus particulièrement aux différentes méthodes d'interpolation (Lagrange, Newton, Hermite, Spline linéaire et Cubique) et de rendre l'erreur entre la fonction et le polynôme le plus petit possible.

**Chapitre 3 :** on va voir des applications très précises sur différentes méthodes d'interpolation par application des codes sous le logiciel Matlab.



# Chapitre 1

## Généralité

Dans ce chapitre, nous rappelons quelques résultats sur les polynômes (ou fonctions polynomiales) et on donne quelques notions dont le besoin dans les autres chapitres. Nous allons également passer à une introduction sur l'interpolation polynomiale.

### 1.1 Espace des polynômes

**Définition 1.1.1** *Un polynôme  $P$  est une fonction  $\mathbb{R} \rightarrow \mathbb{R}$  telle que :*

$$\exists n \in \mathbb{N}, \exists (a_i)_{i=0, \dots, n} \in \mathbb{R}^{n+1}, \forall x \in \mathbb{R}, P(x) = a_0 + a_1x + \dots + a_nx^n \left( = \sum_{i=0}^n a_i x^i \right). \quad (1.1)$$

$P$  est polynôme de degré  $n$  si  $a_n \neq 0$ . Les nombres  $a_i$  s'appellent les coefficients de  $P$ . L'entier non nul  $n$  dans (1.1) est le degré de  $P$  et le coefficient  $a_n$  est le coefficient dominant de  $P$ .

**Définition 1.1.2** *On dit que  $x_0 \in \mathbb{R}$  est racine du polynôme  $P$  ssi  $P(x_0) = 0$ .*

*$x_0$  est racine simple ssi  $P(x_0) = 0$  et  $P'(x_0) \neq 0$ .*

*$x_0 \in \mathbb{R}$  est racine multiple d'ordre  $m$  du polynôme  $P$  ssi  $P(x_0) = 0, P'(x_0) = 0, \dots, P^{(m-1)}(x_0) = 0$  et  $P^{(m)}(x_0) \neq 0$ . Si  $m = 2, 3, 4, \dots$ , on dit racine double, triple, quadruple...*

**Exemple 1.1.1** Si  $P(x) = (x + 1)(x - 3)^2$ , alors  $(-1)$  est racine simple et  $3$  est racine double.

**Proposition 1.1.1** Soit  $P$  un polynôme de degré  $n$ .

- Si  $x_0$  est racine de  $P$ , alors il existe un polynôme  $Q$  de degré  $(n - 1)$  tel que

$$P(x) = (x - x_0)Q(x),$$

et si  $P$  a  $n$  racines simples  $(x_i)_{i=0, \dots, n-1}$ , il est de la forme  $P(x) = k \prod_{i=0}^{n-1} (x - x_i)$ , où  $k$  est une constante.

- Un polynôme  $P$  de degré  $n$  qui a  $(n + 1)$  racines est le polynôme nul.

**Théorème 1.1.1 (de Rolle)** Soit  $f$  une application continue de  $[a, b]$  dans  $\mathbb{R}$  et dérivable sur  $]a, b[$ , avec  $f(a) = f(b)$ . Alors il existe au moins un réel  $c \in ]a, b[$  tel que  $f'(c) = 0$ .

Ce lemme suivant, découle du théorème de Rolle.

**Lemme 1.1.1** Soit  $f$  une fonction  $n$  fois dérivable sur  $[a, b]$ . On suppose qu'il existe  $(n + 1)$  points  $c_0 < c_1 < \dots < c_n$  de  $[a, b]$  tels que  $f(c_i) = 0$ . Alors il existe  $\xi \in ]c_0, c_n[$  tel que  $f^{(n)}(\xi) = 0$ .

**Preuve.** Par récurrence sur  $n$ .

Pour  $n = 1$ , c'est le théorème de Rolle.

Supposons le lemme démontré pour  $(n - 1)$ . Le théorème de Rolle donne des points  $\gamma_0 \in ]c_0, c_1[$ ,  $\dots$ ,  $\gamma_{n-1} \in ]c_{n-1}, c_n[$  tels que  $f'(\gamma_i) = 0$ . Par hypothèse de récurrence, il existe donc  $\xi \in ]\gamma_0, \gamma_{n-1}[ \subset ]c_0, c_n[$  tel que  $(f')^{(n-1)}(\xi) = f^{(n)}(\xi) = 0$ . ■

**Proposition 1.1.2 (Formule de Taylor)**

Soit  $P$  un polynôme qui s'écrit se forme :  $P(x) = \sum_{i=0}^n a_i x^i$ , alors :

$$P(x) = \sum_{i=0}^n \frac{P^{(i)}(x_0)}{i!} (x - x_0)^i,$$

où  $P^{(i)}$  désigne la dérivée  $i$  - ième de  $P$ , et  $x_0$  un réel.

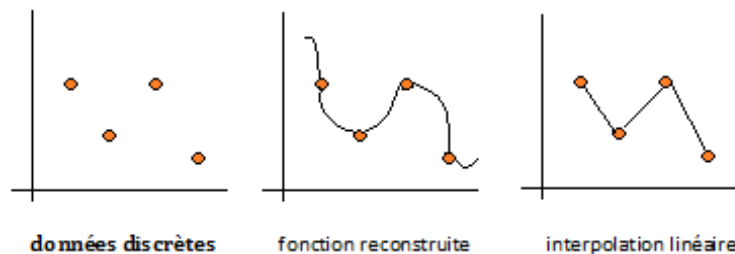
## 1.2 Problème général de l'interpolation polynomaile

Le problème posé est de reconstituer une courbe à partir de la donnée d'un nombre fini de points, ou une fonction à partir de la donnée d'un nombre fini de valeurs.

Parmi les différentes méthodes numériques, il existe une méthode très efficace s'appelle l'interpolation qui permet de répondre à ce problème.

L'interpolation consiste donc à trouver une fonction passant exactement par les points à notre disposition.

L'interpolation a pour fonction l'estimation de données en des points intermédiaires non évalués, en d'autres termes l'interpolation sert à transformer un ensemble de points en une courbe. L'illustration la plus triviale est l'interpolation linéaire où les points de mesure sont reliés par des segments de droites.



L'interpolation polynomiale est un outil pour la construction des méthodes d'intégration numérique ou des méthodes d'approximation des équations différentielles.

### 1.2.1 Position du problème d'interpolation

Supposons qu'on a  $(n + 1)$  couples de valeurs réelles  $((x_i, y_i), i \in \{0, 1, \dots, n\})$  obtenus à travers une mesure expérimentale ou d'une table.

Le but du problème d'interpolation est de déterminer une fonction simple  $F$  qui passe par les points  $(x_i, y_i)$  donnés, c'est-à-dire

$$F(x_i) = y_i, \quad i = 0, \dots, n.$$

La figure (1.1) résume la situation :

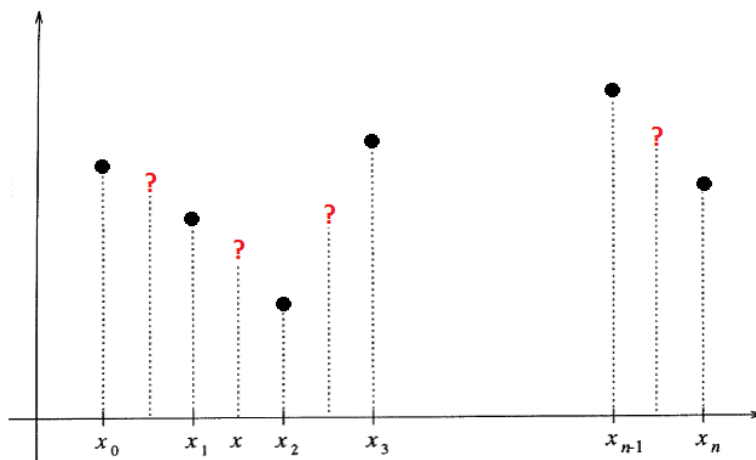


FIG. 1.1 – Problème d'interpolation.

Les points  $(x_i, y_i)$  sont appelés points de collocation ou points d'interpolation. Généralement, on choisit la fonction  $F$  sous forme d'un polynôme, ça ce qu'on appelle l'interpolation polynomiale.

Existe-t-il un polynôme  $P$  tel que  $P(x_i) = y_i, i = 0, \dots, n$  ?

Si oui, quel est son degré ? Est-il unique ? Quelle est l'expression de  $P(x)$  en fonction des données  $(x_i)$  et  $(y_i)$  ?

### Existence et unicité du problème d'interpolation

**Proposition 1.2.1** *Soient  $(n + 1)$  couples de valeurs réelles  $((x_i, y_i), i \in \{0, 1, \dots, n\})$ , et si les  $x_i$  sont tous distincts alors il existe un unique polynôme  $P_n$  de degré inférieur ou égal à  $n$  vérifiant*

$$P_n(x_i) = y_i, \quad i = 0, \dots, n.$$

**Preuve. 1/ Unicité :** *On utilise le raisonnement par l'absurde, supposons qu'il existe*



Ce système possède une unique solution si et seulement si la matrice carrée qui lui est associée est inversible. Or, il se trouve que cette dernière est une matrice de Vandermonde dont le déterminant vaut :

$$\begin{vmatrix} 1 & x_0 & x_0^2 & x_0^3 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & x_1^3 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & x_2^3 & \dots & x_2^n \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & x_n^3 & \dots & x_n^n \end{vmatrix} = \prod_{0 \leq i < j \leq n} (x_j - x_i) = \prod_{i=0}^{n-1} \left( \prod_{j=i+1}^n (x_j - x_i) \right).$$

Les nœuds d'interpolation étant tous distincts, ce déterminant est non nul. Donc, nous sommes assurés de l'existence et unicité de la solution. ■

**Exemple 1.2.1** Notre but est de rechercher sur un polynôme qui passe exactement par les 3 points distincts suivantes :  $(1, 1)$ ,  $(0, 1)$  et  $(2, -1)$ . Le polynôme recherché est tout au plus de degré 2. Donc, il vérifie :

$$\begin{aligned} P_2(x_i) &= y_i \text{ pour } i = 0, 1, 2. \\ \iff a_0 + a_1x_i + a_2x_i^2 &= y_i, \\ \iff \begin{cases} a_0 + a_1x_0 + a_2x_0^2 = y_0 \\ a_0 + a_1x_1 + a_2x_1^2 = y_1 \\ a_0 + a_1x_2 + a_2x_2^2 = y_2 \end{cases} \end{aligned}$$

Qui s'écrit sous forme matricielle :

$$\underbrace{\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 2 & 4 \end{pmatrix}}_A \underbrace{\begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}}_X = \underbrace{\begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}}_B \implies X = A^{-1}B \text{ telque : } A^{-1} = \frac{1}{\det A} (\text{Com}A)^t$$

- En calculons d'abord  $A^{-1}$  :

$$\det A = \begin{vmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 2 & 4 \end{vmatrix} = - \begin{vmatrix} 1 & 1 \\ 2 & 4 \end{vmatrix} = -(4 - 2) = -2 \neq 0 \text{ donc : } A \text{ est inversible.}$$

$$\text{Com}A = \begin{pmatrix} 0 & -4 & 2 \\ -2 & 3 & -1 \\ 0 & 1 & -1 \end{pmatrix} \Rightarrow (\text{Com}A)^t = \begin{pmatrix} 0 & -2 & 0 \\ -4 & 3 & 1 \\ 2 & -1 & -1 \end{pmatrix}$$

$$\text{Alors : } A^{-1} = \frac{1}{\det A} (\text{Com}A)^t = \frac{-1}{2} \begin{pmatrix} 0 & -2 & 0 \\ -4 & 3 & 1 \\ 2 & -1 & -1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 2 & \frac{-3}{2} & \frac{-1}{2} \\ -1 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

- Maintenant en calculons  $X$  :

$$X = A^{-1}B = \begin{pmatrix} 0 & 1 & 0 \\ 2 & \frac{-3}{2} & \frac{-1}{2} \\ -1 & \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} \iff \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}$$

Le polynôme recherché est donc :

$$P_2(x) = 1 + x - x^2.$$

# Chapitre 2

## Méthodes d'interpolation

Après avoir confirmé l'existence et l'unicité de la solution du problème d'interpolation à l'aide du matrice de Vandermonde. Nous continuons à apprendre sur les différentes méthodes d'interpolation qui contribueraient à l'enrichissement de ce chapitre.

### 2.1 Méthodes d'interpolation

#### 2.1.1 Interpolation de Lagrange (1736 – 1813)

Si l'on se situe dans un contexte historique, à l'époque de Lagrange, le calcul des coefficients ne pouvait pas s'effectuer par la résolution d'un système linéaire dès que l'ordre de la matrice dépassait quelques unités, et Lagrange proposa une technique qui a permis de réaliser directement l'interpolation sans calculer explicitement les coefficients du polynôme. C'est cette méthode que nous exposons maintenant.

On se donne  $(n+1)$  points de collocation  $(x_i)_{i=0,\dots,n}$  distincts 2 à 2, et  $(n+1)$  réels  $(y_i)_{i=0,\dots,n}$ . Et on cherche un polynôme  $P$  de degré  $n$  tel que  $P(x_i) = y_i$  pour tout  $i$ .

#### Cas linéaire

On prendre deux points expérimentaux  $(x_0, y_0)$  et  $(x_1, y_1)$ . Soit  $P_1(x)$ , l'unique polynôme d'ordre 1 interpolant ces points. Il peut se mettre sous la forme  $P_1(x) = ax + b$ . Comme



il passe par les deux points  $(x_0, y_0)$  et  $(x_1, y_1)$ , les coefficients  $a$  et  $b$  doivent satisfaire :

$$\begin{cases} ax_0 + b = y_0 & (1) \\ ax_1 + b = y_1 & (2) \end{cases}$$

Pour résoudre ce système on passe par :

$$\begin{aligned} (1) &\implies b = y_0 - ax_0 \\ (1) - (2) &\implies ax_0 + b - ax_1 - b = y_0 - y_1 \\ &\implies a(x_0 - x_1) = y_0 - y_1 \\ &\implies a = \frac{y_0 - y_1}{x_0 - x_1}, \end{aligned}$$

donc

$$\begin{aligned} P_1(x) &= ax + b \\ &= ax + (y_0 - ax_0) \\ &= a(x - x_0) + y_0 \\ &= \frac{y_0 - y_1}{x_0 - x_1}(x - x_0) + y_0 \\ &= y_0 \underbrace{\frac{x - x_1}{x_0 - x_1}}_{L_0(x)} + y_1 \underbrace{\frac{x_0 - x}{x_0 - x_1}}_{L_1(x)}. \end{aligned}$$

On remarque que  $L_0(x_0) = 1$ ,  $L_0(x_1) = 0$  et  $L_1(x_0) = 0$ ,  $L_1(x_1) = 1$ .

Le polynôme de degré 1 est donc :

$$P_1(x) = y_0 L_0(x) + y_1 L_1(x).$$

Ces deux fonctions sont à leur tour illustrées à la figure (2.1) :

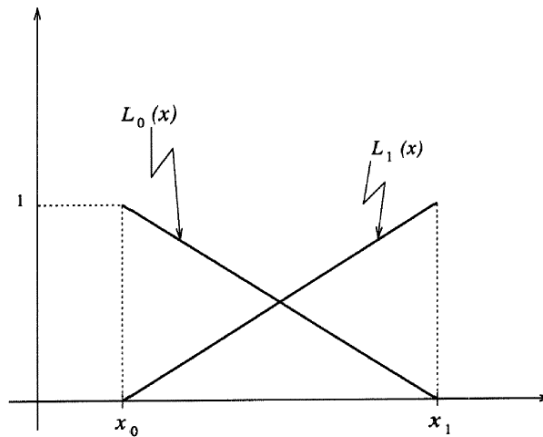


FIG. 2.1 – Polynômes de Lagrange de degré 1 :  $L_0(x)$  et  $L_1(x)$ .

**Exemple 2.1.1** *L'équation de la droite passant par les points  $(0, 1)$  et  $(2, 2)$  est :*

$$\begin{aligned}
 P_1(x) &= y_0 L_0(x) + y_1 L_1(x) \\
 &= y_0 \frac{x - x_1}{x_0 - x_1} + y_1 \frac{x_0 - x}{x_0 - x_1} \\
 &= 1 \frac{(x - 2)}{(0 - 2)} + 2 \frac{(0 - x)}{(0 - 2)} \\
 &= \frac{1}{2}x + 1.
 \end{aligned}$$

### Cas parabolique

On prend de trois points expérimentaux  $(x_0, y_0)$ ,  $(x_1, y_1)$  et  $(x_2, y_2)$ . On recherche le polynôme d'ordre 2,  $P_2(x)$  interpolant ces points :

$$P_2(x) = ax^2 + bx + c.$$

Les coefficients doivent satisfaire :

$$\begin{cases}
 ax_0^2 + bx_0 + c = y_0 \\
 ax_1^2 + bx_1 + c = y_1 \\
 ax_2^2 + bx_2 + c = y_2
 \end{cases}$$

On démontre alors facilement que

$$P_2(x) = y_0 \underbrace{\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}}_{L_0(x)} + y_1 \underbrace{\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}}_{L_1(x)} + y_2 \underbrace{\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}}_{L_2(x)}.$$

Ces trois fonctions sont à leur tous illustrées à la figure (2.2) :

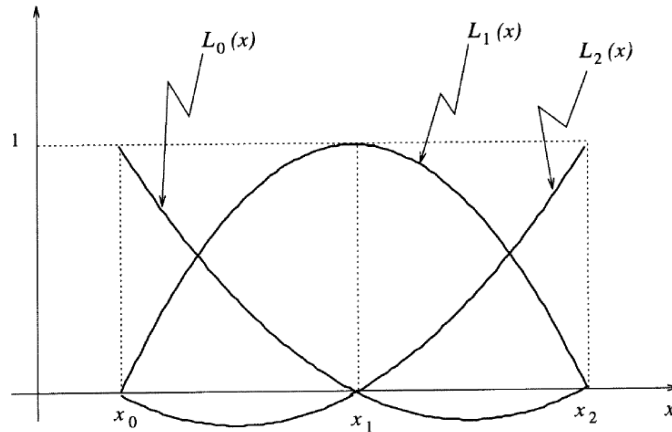


FIG. 2.2 – Polynôme de Lagrange de degré 2 :  $L_0(x)$ ,  $L_1(x)$  et  $L_2(x)$ .

Le polynôme de degré 2 est donc :

$$P_2(x) = y_0 L_0(x) + y_1 L_1(x) + y_2 L_2(x).$$

**Exemple 2.1.2** La parabole passant par les points  $(2, -1)$ ,  $(0, 1)$  et  $(1, 1)$  est donnée par :

$$\begin{aligned} P_2(x) &= y_0 L_0(x) + y_1 L_1(x) + y_2 L_2(x) \\ &= y_0 \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + y_1 \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + y_2 \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} \\ &= (-1) \frac{(x-0)(x-1)}{(2-0)(2-1)} + \frac{(x-2)(x-1)}{(0-2)(0-1)} + \frac{(x-2)(x-0)}{(1-2)(1-0)} \\ &= -x^2 + x + 1. \end{aligned}$$

**Cas de  $(n+1)$  points**

On analyse le cas général de la même façon, on trouve que le polynôme  $P_n(x)$  est de la forme :

$$P_n(x) = \sum_{i=0}^n y_i L_i(x),$$

où :

$$L_i(x) = \frac{(x - x_0) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)}{(x_i - x_0) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)} \quad \text{pour } i = \overline{0, n} \text{ et } x \in \mathbb{R},$$

tels que :

$$L_i(x_i) = 1 \quad \forall i, \quad L_i(x_j) = 0 \quad \forall i \neq j.$$

$L_i(x)$  est appelé : Polynôme de Lagrange aux points  $x_i$ .

$P_n(x)$  s'appelle : Le polynôme d'interpolation de Lagrange.

**Théorème 2.1.1** Soient  $\{(x_i, y_i) \text{ pour } i = 0, 1, \dots, n\}$   $(n + 1)$  points d'interpolation données, tels que les  $x_i$  soient tous deux à deux différents. L'unique polynôme d'interpolation de degré  $n$  passant par tous ces points peut s'écrire :

$$P_n(x) = \sum_{i=0}^n y_i L_i(x), \tag{2.1}$$

où les  $(n + 1)$  fonctions  $L_i(x)$  sont définies par la relation (ci-dessus). C'est la formule de Lagrange.

**Remarque 2.1.1** Le principal inconvénient de la méthode d'interpolation de Lagrange est que le fait de rajouter un nœud change complètement les interpolants de base de Lagrange et on doit donc recalculer entièrement le polynôme  $P_n(x)$ . On va donc considérer une autre approche qui se comporte mieux lorsqu'on rajoute des nœuds, ce que nous appelons *l'interpolation de Newton*.

### 2.1.2 Interpolation de Newton (1643 – 1727)

Le but principal de cette formule est de déterminer les  $(n+1)$  coefficients  $a_i$  d'un polynôme  $P_n(x)$  qui passe par les  $(n+1)$  points de collocation  $(x_i, f(x_i))$  qui vérifient :

$$P_n(x_i) = f(x_i) \text{ pour } i = 0, 1, \dots, n.$$

L'idée de Newton est d'écrire les polynômes de manière différente : à partir du polynôme constant (de degré 0) :

$$P_0(x) = a_0,$$

ce qui revient à se donner un point  $x_0$  et la valeur  $a_0 = f(x_0)$ , c'est le premier coefficient. On ajoute un point  $(x_1, f(x_1)) \in \mathbb{R}^2$  et on veut obtenir le polynôme de degré 1 sous la forme :

$$P_1(x) = a_0 + a_1(x - x_0) = P_0(x) + a_1(x - x_0).$$

Comme on doit avoir  $P_1(x_1) = f(x_1)$ , il vient :  $a_1 = \frac{f(x_1) - f(x_0)}{(x_1 - x_0)}$ , c'est le deuxième coefficient.

Puis on ajoute un point  $(x_2, f(x_2))$  pour obtenir :

$$P_2(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) = P_1(x) + a_2(x - x_0)(x - x_1).$$

Comme on doit avoir  $P_2(x_2) = f(x_2)$ , il vient :

$$\begin{aligned} f(x_2) &= a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) \\ &= f(x_0) + \left( \frac{f(x_1) - f(x_0)}{(x_1 - x_0)} \right) (x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1), \end{aligned}$$

alors :

$$\begin{aligned}
 a_2 &= \frac{1}{(x_2 - x_0)(x_2 - x_1)} \left[ f(x_2) - f(x_0) - \left( \frac{f(x_1) - f(x_0)}{x_1 - x_0} \right) (x_2 - x_0) \right] \\
 &= \frac{1}{(x_2 - x_0)} \left[ \frac{f(x_2) - f(x_1) + f(x_1) - f(x_0)}{x_2 - x_1} - \left( \frac{f(x_1) - f(x_0)}{x_1 - x_0} \right) \frac{(x_2 - x_0)}{x_2 - x_1} \right] \\
 &= \frac{1}{(x_2 - x_0)} \left[ \frac{f(x_2) - f(x_1)}{x_2 - x_1} + \left( \frac{f(x_1) - f(x_0)}{x_1 - x_0} \right) \frac{(x_1 - x_0)}{x_2 - x_1} - \left( \frac{f(x_1) - f(x_0)}{x_1 - x_0} \right) \frac{(x_2 - x_0)}{x_2 - x_1} \right] \\
 &= \frac{1}{(x_2 - x_0)} \left[ \frac{f(x_2) - f(x_1)}{x_2 - x_1} - \left( \frac{f(x_1) - f(x_0)}{x_1 - x_0} \right) \frac{(x_2 - x_1)}{x_2 - x_1} \right] \\
 &= \frac{1}{(x_2 - x_0)} \left[ \frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0} \right].
 \end{aligned}$$

On poursuit le processus : disposant des points  $(x_i, f(x_i))$  pour  $i = 0, \dots, k - 1$ , on ajoute un point  $(x_k, f(x_k))$  et le polynôme obtenu s'écrit :

$$P_k(x) = P_{k-1}(x) + a_k(x - x_0)(x - x_1)\dots(x - x_{k-1}),$$

et donc  $P_k - P_{k-1}$  est le polynôme de degré  $k$  qui s'annule aux  $k$  points  $x_0, \dots, x_{k-1}$  et qui vaut  $f(x_k)$  au point  $x_k$ , ce qui détermine  $a_k$ .

On remarque que si on ajoute plus de deux points les formules deviennent plus compliquées, ça ce que demande d'introduire la notion de **différences divisées**.

**Définition 2.1.1** (*Différences divisées*)

Soit  $\{(x_i, f(x_i))\}_{i=0}^n$  un ensemble de  $(n + 1)$  points distincts.

Les premières différences divisées de la fonction  $f(x)$  définie en deux points  $x_i$  et  $x_{i+1}$  sont :

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

Les deuxièmes différences divisées de la fonction  $f(x)$  définie en trois points  $x_i, x_{i+1}$  et  $x_{i+2}$  sont :

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

De même, les  $n^{es}$  différences divisées de la fonction  $f(x)$  sont définies à partir des  $(n - 1)^{es}$

différences divisées de la façon suivante :

$$f[x_0, x_1, x_2, \dots, x_n] = \frac{f[x_1, x_2, \dots, x_n] - f[x_0, x_1, x_2, \dots, x_{n-1}]}{x_n - x_0}.$$

**Notation 2.1.1** D'après la définition précédente, on peut résumer la suite :

Le coefficient  $a_1$  s'écrit se la forme :  $a_1 = f[x_0, x_1]$ , et le polynôme de degré 1 est :

$$P_1(x) = f(x_0) + f[x_0, x_1](x - x_0).$$

Le coefficient  $a_2$  s'écrit se la forme :  $a_2 = f[x_0, x_1, x_2]$ , et le polynôme de degré 2 est :

$$P_2(x) = f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1).$$

Le coefficient  $a_i$  s'écrit se la forme :  $a_i = f[x_0, x_1, x_2, \dots, x_i]$  pour  $0 \leq i \leq n$ , et le polynôme de degré  $i$  est :

$$P_i(x) = f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \dots + f[x_0, x_1, x_2, \dots, x_i](x - x_0)(x - x_1) \dots (x - x_{i-1}).$$

**Remarque 2.1.2** Pour expliciter le processus récursif, les différences divisées peuvent être calculées en les disposant de la manière suivante dans un tableau :

$x_i$	$f(x_i)$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$\cdots$	$f[x_i, x_{i+1}, x_{i+2}, \dots, x_n]$
$x_0$	$f(x_0)$				
$x_1$	$f(x_1)$	$f[x_0, x_1]$			
$x_2$	$f(x_2)$	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$		
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	
$x_n$	$f(x_n)$	$f[x_{n-1}, x_n]$	$f[x_{n-2}, x_{n-1}, x_n]$	$\cdots$	$f[x_0, x_1, x_2, \dots, x_n]$

(2.2)

On en déduit immédiatement un algorithme de calcul des différences divisées. On note que contrairement à ce qu'il se passait pour l'interpolation de Lagrange, l'ajout d'un nouveau nœud n'oblige pas à recalculer toutes les différences divisées. Plus précisément, dans chaque numérateur, les  $(k - 1)$  premiers arguments du premier terme sont identiques aux  $(k - 1)$  derniers arguments du deuxième terme et le dénominateur est la différence entre les arguments qui diffèrent d'un terme à l'autre. Nous en déduisons que  $f[x_0, x_1, x_2, \dots, x_k]$  peut s'exprimer comme le quotient de la différence de deux différences divisées d'ordre  $(k - 1)$  (ayant  $(k - 1)$  arguments en commun) par la différence entre arguments différents :

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

**Théorème 2.1.2 (Formule de Newton 1669)**

Soient  $\{(x_i, f(x_i))$  pour  $i = 0, 1, \dots, n\}$   $(n + 1)$  points d'interpolation données, où les  $x_i$  sont distincts. L'unique polynôme d'interpolation de degré  $n$  qui passe par tous ces points et donné par :

$$\begin{aligned} P_n(x) = & f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\ & + \dots + f[x_0, x_1, x_2, \dots, x_n](x - x_0)(x - x_1) \dots (x - x_{n-1}). \end{aligned} \quad (2.3)$$

**Preuve.** Démonstration par récurrence. C'est vrai pour  $P_1$ .

Supposons que ce le soit pour  $P_{n-1}$ ,  $n \geq 2$ , i.e. qu'étant donnés  $n$  points,  $(x_i, y_i)_{i=0, \overline{n-1}}$  où  $y_i = f(x_i)$ , le polynôme  $P_{n-1}$  de Newton est donné par :

$$P_{n-1}(x) = f(x_0) + f[x_0, x_1](x - x_0) + \dots + f[x_0, \dots, x_{n-1}](x - x_0) \dots (x - x_{n-2}).$$

Et pour les points  $(x_i, f(x_i))_{i=1, \overline{n}}$ , le polynôme  $q_{n-1}$  de Newton est donné par :

$$q_{n-1}(x) = f(x_1) + f[x_1, x_2](x - x_1) + \dots + f[x_1, \dots, x_n](x - x_1) \dots (x - x_{n-1}).$$



Et par hypothèse de récurrence :

$$P_{n-1}(x_i) = q_{n-1}(x_i) \quad (= y_i = f(x_i)) \quad \text{pour } i = 1, \dots, n-1.$$

D'où immédiatement, en posant :

$$P_n(x) = \frac{(x_n - x)P_{n-1}(x) + (x - x_0)q_{n-1}(x)}{(x_n - x_0)}, \quad (2.4)$$

on a  $P_n(x_i) = f(x_i)$  pour tout  $i = 0, \dots, n$ . Donc  $P_n$  est le polynôme cherché. Il reste à montrer que :

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

Mais posant :

$$S_n(x) = P_n(x) - P_{n-1}(x) = \frac{(x - x_0)}{(x_n - x_0)} (q_{n-1}(x) - P_{n-1}(x)),$$

le polynôme  $S_n$  de degré  $n$  s'annule en les  $n$  points  $x_i, 0 \leq i \leq n-1$  : c'est trivial pour  $x = x_0$  et on a  $P_{n-1}(x_i) = q_{n-1}(x_i)$  pour  $1 \leq i \leq n-1$ . Donc  $S_n$  est de la forme  $S_n(x) = \alpha(x - x_0) \dots (x - x_{n-1})$  pour un scalaire  $\alpha$  donné. Et il reste à montrer que  $\alpha = f[x_0, x_1, x_2, \dots, x_n]$ .

Mais le coefficient de  $x^n$  dans  $P_n$  est donné à l'aide de (2.4) par :

$$\frac{-f[x_0, x_1, \dots, x_{n-1}] + f[x_1, x_2, \dots, x_n]}{(x_n - x_0)} \quad \text{donc } \alpha = f[x_0, x_1, \dots, x_n],$$

par définition de  $f[x_0, x_1, \dots, x_n]$ . ■

**Exemple 2.1.3** • On veut calculer le polynôme d'interpolation par la méthode de Newton en les points de collocations suivants :  $(1, -1)$ ,  $(0, 1)$  et  $(2, 1)$ .

On cherche donc  $P_2 \in \mathbb{R}_2[X]$  tel que  $P_2(x_i) = f(x_i)$  pour  $i = 0, 1, 2$ .

On construira le tableau des différences divisées :

$i$	$x_i$	$f(x_i)$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$
0	1	-1		
1	0	1	-2	
2	2	1	0	2

On a alors :

$$\begin{aligned}
 P_2(x) &= f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\
 &= -1 - 2(x - 1) + 2x(x - 1) \\
 &= 2x^2 - 4x + 1.
 \end{aligned}$$

- On ajout un nouveau point de collocation  $(3, 0)$ , en déduite donc le polynôme d'interpolation  $P_3$  :

$$\begin{aligned}
 P_3(x) &= P_2(x) + f[x_0, x_1, x_2, x_3](x - x_0)(x - x_1)(x - x_2) \\
 &= 2x^2 - 4x + 1 - \frac{7}{6}x(x - 1)(x - 2) \\
 &= -\frac{7}{6}x^3 + \frac{11}{2}x^2 - \frac{19}{3}x + 1.
 \end{aligned}$$

### 2.1.3 Interpolation d'Hermite ou Osculateur (1822 – 1901)

Notre but consiste à construire sur un polynôme qui interpole la fonction  $f$  ainsi que sa dérivée  $f'$  aux points  $(x_i)_{0 \leq i \leq n}$  d'un intervalle  $[a, b]$ . Charles Hermite (1822 – 1901) a généralisé l'interpolation de Lagrange en faisant coïncider non seulement  $f$  et  $P_n$  aux points  $x_i$ , mais aussi leurs dérivées d'ordre  $k_i$  aux points  $x_i$ .

Soient  $x_0, x_1, \dots, x_i, (n + 1)$  abscisses distinctes, et  $f$  une fonction dont on connaît les

valeurs  $f(x_i)$  et  $f'(x_i)$  pour  $i = 0, 1, \dots, n$ . On cherche un polynôme  $P$  tel que

$$\begin{cases} P(x_i) = f_i & \text{pour } i = \overline{0, n}. \\ P'(x_i) = f'_i & \text{pour } i = \overline{0, n}. \end{cases} \quad (2.5)$$

**Théorème 2.1.3** *Il existe un unique  $P \in \mathbb{R}_{2n+1}[X]$  satisfaisant (2.5).*

**Preuve. 1/ Unicité :** Supposons qu'il existe deux polynômes  $p$  et  $q$  de degré au plus  $(2n + 1)$  vérifiant (2.5) c-à-d :  $p(x_i) = q(x_i) = f_i$  et  $p'(x_i) = q'(x_i) = f'_i$  pour  $i = 0, \dots, n$ . Le polynôme  $R = p - q$ , qui est aussi de degré au plus  $(2n + 1)$ , admet chaque  $x_i$  comme racine double ( $R(x_i) = R'(x_i) = 0$ ). Il possède donc  $(2n + 2)$  racines au moins,  $R$  est donc le polynôme nul.

**2/ Existence :** Base de polynômes d'Hermite.

On cherche une base de polynômes de  $\mathbb{R}_{2n+1}[X]$  telle que

$$P(x) = \sum_{i=0}^n f_i A_i(x) + \sum_{i=0}^n f'_i B_i(x)$$

Les conditions sur les fonctions de bases sont alors les suivantes :

$$\begin{aligned} A_i(x_j) &= \delta_{ij}, \quad B_i(x_j) = 0 \text{ pour } i = \overline{0, n} \\ A'_i(x_j) &= 0, \quad B'_i(x_j) = \delta_{ij} \text{ pour } i = \overline{0, n} \end{aligned}$$

les premières conditions permettent d'imposer  $P(x_i) = f_i$  et les secondes  $P'(x_i) = f'_i$ .

Ces conditions permettent de construire les fonctions de bases. En effet,

Construction des polynômes  $A_i$ . On a  $A_i(x_j) = A'_i(x_j) = 0$  pour  $i \neq j$  alors  $(x - x_j)^2$  divise  $A_i$  pour  $i \neq j$ , alors  $A_i(x) = R(x) \prod_{j=0, j \neq i}^n (x - x_j)^2$  où  $R(x) \in \mathcal{P}_{N1}$ . On peut exprimer ce polynôme en fonction du polynôme de Lagrange. En effet,  $L_i(x) = \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j}$ , ainsi

$A_i(x) = q(x)L_i^2(x)$ , où  $q(x) = ax + b \in P_1$ . Les coefficients  $a$  et  $b$  sont tels que

$$A_i(x_i) = 1 \text{ et } A_i'(x_i) = 0.$$

On a

$$A_i(x_i) = (ax_i + b) L_i^2(x_i) = 1 = ax_i + b \text{ car } L_i(x_i) = 1.$$

$$A_i'(x_i) = aL_i^2(x_i) + 2L_i(x_i)L_i'(x_i)(ax_i + b) = a + 2L_i'(x_i)(ax_i + b) = a + 2L_i'(x_i) = 0,$$

ainsi  $a = -2L_i'(x_i)$  et  $b = 1 - ax_i$ , enfin

$$A_i(x) = (1 - 2(x - x_i) L_i'(x_i))L_i^2(x_i).$$

Calcul de  $B_i$ . Pour  $j \neq i$ , on a  $B_i(x_j) = B_i'(x_j) = 0$ , alors  $L_j^2$  divise  $B_i$ , d'autre part  $B_i(x_i) = 0$ , alors  $(x - x_i)$  divise aussi  $B_i$ . On déduit que  $B_i(x) = c(x - x_i)L_i^2(x)$  et  $c \in \mathbb{R}$ .

On détermine la constante par la relation  $B_i'(x_i) = 1$ ; on a

$$B_i'(x_i) = cL_i^2(x_i) + 2c(x_i - x_i)L_i(x_i)L_i'(x_i) = c = 1,$$

ce qui donne

$$B_i(x) = (x - x_i)L_i^2(x).$$

■

**Définition 2.1.2** On appelle un polynôme d'interpolation d'Hermite le polynôme de la forme :

$$P_{2n+1}(x) = \sum_{i=0}^n f(x_i)H_i(x) + \sum_{i=0}^n f'(x_i)V_i(x), \quad (2.6)$$

avec :  $H_i(x) = [1 - 2(x - x_i)L_i'(x_i)]L_i^2(x)$ ,  $V_i(x) = (x - x_i)L_i^2(x)$ ,

où les  $H_i$  et  $V_i$  sont des polynômes de degré  $(2n + 1)$ , et

$$L_i(x) = \frac{\prod_{\substack{j=0 \\ j \neq i}}^n (x - x_j)}{\prod_{\substack{j=0 \\ j \neq i}}^n (x_i - x_j)}, \text{ polynôme de Lagrange au point } x_i.$$

**Remarque 2.1.3** *L'interpolation de Lagrange est un cas particulier de l'interpolation d'Hermite.*

**Exemple 2.1.4** *On veut calculer le polynôme d'interpolation par la méthode d'Hermite tel que*

$$\begin{cases} P(0) = 1, & P(1) = 2 \\ P'(0) = 0, & P'(1) = 0 \end{cases}$$

On a :

$$\begin{aligned} P_3(x) &= \sum_{i=0}^1 f(x_i)H_i(x) + \sum_{i=0}^1 f'(x_i)V_i(x) \\ &= f(x_0)[1 - 2(x - x_0)L'_0(x_0)]L_0^2(x) + f'(x_0)(x - x_0)L_0^2(x) \\ &\quad + f(x_1)[1 - 2(x - x_1)L'_1(x_1)]L_1^2(x) + f'(x_1)(x - x_1)L_1^2(x) \\ &= (1 + 2x)(x - 1)^2 + 2(1 - 2(x - 1))x^2 \\ &= -2x^3 + 3x^2 + 1. \end{aligned}$$

Avec :

$$\begin{aligned} L_0(x) &= \frac{x - x_1}{x_0 - x_1} = -(x - 1), & L'_0(x_0) &= (-1), & L_0^2(x) &= (x - 1)^2, \\ L_1(x) &= \frac{x - x_0}{x_1 - x_0} = x, & L'_1(x_1) &= 1, & L_1^2(x) &= x^2. \end{aligned}$$

### 2.1.4 Interpolation par les fonctions splines

L'interpolation polynomiale possède deux défauts majeurs inévitables :

- Le coût des calculs devient élevé lorsque le degré du polynôme est grand.

- Les effets de bord sont importants si l'intervalle d'interpolation est grand.

Une solution consiste à travailler localement : on fait alors de l'interpolation par morceaux, c'est-à-dire que sur un sous intervalle donné on fait de l'interpolation polynomiale de degré faible, puis on impose des conditions de régularité (continuité ou plus) pour relier les morceaux.

### Interpolation par spline linéaire

Pour répartition, non nécessairement uniforme, de points  $a = x_0 < x_1 < \dots < x_n = b$ , on approxime la fonction  $f$  sur chaque sous-intervalle  $[x_i, x_{i+1}]$ , par un segment de droite connectant les deux points  $(x_i, f(x_i))$  et  $(x_{i+1}, f(x_{i+1}))$ . Sur chaque sous-intervalle, ce segment de droite s'écrit :

$$P_i(x) = f(x_i) + \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}(x - x_i), \text{ pour } x_i \leq x \leq x_{i+1}. \quad (2.7)$$

### Interpolation par spline cubique

Nous avons déjà, dans les sections précédentes, montré que l'interpolation polynomiale n'est pas bien adaptée à l'approximation de fonctions pour des degrés  $n$  élevés. Pour ça on procède à une interpolation polynomiale par morceaux. L'avantage de ce type d'interpolation est qu'en augmentant le degré  $n$ , on accroît simultanément le nombre de morceaux et non le degré du polynôme. Les fonctions splines d'interpolation cubiques sont les fonctions splines de plus petit degré interpolant une fonction  $f$  régulière, deux fois dérivable sur  $[a, b]$ . La régularité d'une fonction peut se mesurer au moyen de ses dérivées. Ainsi, plus une fonction est différentiable, plus la courbe qui lui est associée est lisse et plus la fonction est régulière.

Les fonctions spline ne sont plus des polynômes, elles sont seulement polynomiale par morceaux. Elles sont caractérisées par les trois propriétés :

1. polynomiales par morceaux, de degré trois (d'où le terme cubique) sur chaque  $[x_i, x_{i+1}]$ .

2. elles sont de classe  $C^2$ . En chaque  $x_i$ , les dérivées premières et secondes à gauche et à droite coïncident.

3. la dérivée seconde s'annule aux deux extrémités  $x_0$  et  $x_n$ .

Afin de réaliser cette interpolation, on divise l'intervalle  $[a, b]$  en  $n$  sous-intervalles de taille  $(b - a) / n$ . Et à partir des propriétés 1, 2, 3 il est facile de calculer la fonction  $P$ , et sur chaque sous-intervalle  $[x_i, x_{i+1}]$  on prend  $h = x_{i+1} - x_i$ , et on construit un polynôme cubique  $P_i(x)$  d'interpolation. Ce dernier connecte le point  $(x_i, f(x_i))$  au point  $(x_{i+1}, f(x_{i+1}))$  et s'écrit :

$$P_i(x) = f_i + f'_i(x - x_i) + \frac{f''_i}{2!}(x - x_i)^2 + \frac{f'''_i}{3!}(x - x_i)^3, \quad i = 0, 1, \dots, n - 1.$$

Les coefficients à déterminer sont les  $f_i, f'_i, f''_i$  et  $f'''_i$ . Dans le cas de splines not a knot (pas constant),  $P_i(x)$  est évalué par :

$$P_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i. \quad (2.8)$$

On obtient :

$$\begin{aligned} P_i(x_i) &= d_i &= f(x_i) \\ P_i(x_{i+1}) &= a_i h^3 + b_i h^2 + c_i h + d_i = f(x_{i+1}) \\ P'_i(x_i) &= c_i \\ P'_i(x_{i+1}) &= 3a_i h^2 + 2b_i h + c_i \\ P''_i(x_i) &= 2b_i &= f''(x_i) \\ P''_i(x_{i+1}) &= 6a_i h + 2b_i &= f''(x_{i+1}) \end{aligned}$$

On prendre  $\alpha_i = P_i''(x_i)$ , on trouve que :

$$\begin{cases} a_i = & \frac{1}{6h} (\alpha_{i+1} - \alpha_i) \\ b_i = & \frac{1}{2}\alpha_i \\ c_i = & \frac{1}{h} (f(x_{i+1}) - f(x_i)) - \frac{h}{6} (2\alpha_i + \alpha_{i+1}) \\ d_i = & f(x_i) \end{cases}$$

Par l'utilisation de la continuité de la dérivée première, on obtient :

$$P_i'(x_{i+1}) = \frac{1}{h} (f(x_{i+1}) - f(x_i)) + \frac{h}{6} (2\alpha_{i+1} + \alpha_i)$$

En faisant  $i \leftarrow i - 1$ , On obtient la condition de continuité  $P_{i-1}'(x_i) = P_i'(x_i)$

$$\frac{1}{h} (f(x_i) - f(x_{i-1})) + \frac{h}{6} (2\alpha_i + \alpha_{i-1}) = \frac{1}{h} (f(x_{i+1}) - f(x_i)) - \frac{h}{6} (\alpha_{i+1} + 2\alpha_i)$$

En exprimant cette condition en  $x_1, x_2, \dots, x_{n-1}$  on obtient un système linéaire en les inconnues  $\alpha_1, \alpha_2, \dots, \alpha_{n-1}$  dont la  $i^e$  ligne est :

$$h(\alpha_{i-1} + 4\alpha_i + \alpha_{i+1}) = \frac{6}{h}(f(x_{i+1}) - 2f(x_i) + f(x_{i-1}))$$

Il faut rajouter 2 équations afin de compléter le système, en posant la condition  $\alpha_0 = \alpha_n = 0$ . Dans ce cas,  $\alpha = (\alpha_1, \dots, \alpha_{n-1})^T$  est la solution d'un système linéaire, tridiagonale :

$$A\alpha = b_n.$$

Avec :



$$A = h \begin{pmatrix} 4 & 1 & 0 & \cdots & 0 \\ 1 & 4 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & 4 & 1 \\ 0 & \cdots & 0 & 1 & 4 \end{pmatrix} \quad \text{et} \quad b_n = \frac{6}{h} \begin{pmatrix} f(x_0) - 2f(x_1) + f(x_2) \\ f(x_1) - 2f(x_2) + f(x_3) \\ f(x_2) - 2f(x_3) + f(x_4) \\ \vdots \\ f(x_{n-2}) - 2f(x_{n-1}) + f(x_n) \end{pmatrix},$$

avec,  $\alpha$  est un vecteur de dimension  $n - 1$ .

**Exemple 2.1.5** Soit les 3 points suivants  $(0, 1), (1, 1), (2, 5)$ . Interpolons ces données par la fonction spline cubique naturelle et  $h = 1$  et pour tout  $x_i \leq x \leq x_{i+1}$  :

$$\begin{aligned} P_i(x) &= a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \\ &= \frac{1}{6h} (\alpha_{i+1} - \alpha_i) (x - x_i)^3 + \frac{1}{2} \alpha_i (x - x_i)^2 + \frac{1}{h} (f(x_{i+1}) - f(x_i)) (x - x_i) \\ &\quad - \frac{h}{6} (2\alpha_i + \alpha_{i+1}) (x - x_i) + f(x_i). \end{aligned}$$

$$\alpha_0 = \alpha_2 = 0, \alpha_1 = ?$$

$$h (\alpha_{i-1} + 4\alpha_i + \alpha_{i+1}) = \frac{6}{h} (f(x_{i+1}) - 2f(x_i) + f(x_{i-1})), \text{ et } \alpha_i = f''(x_i),$$

$$\text{pour } i = 1 : \quad \alpha_0 + 4\alpha_1 + \alpha_2 = 6(f(x_2) - 2f(x_1) + f(x_0)) \Rightarrow 4\alpha_1 = 6(5 - 2 + 1)$$

$$\Rightarrow \alpha_1 = 6.$$

$$\alpha_0 = 0, \alpha_1 = 6, \alpha_2 = 0.$$

$$\bullet [x_i, x_{i+1}] = [x_0, x_1] = [0, 1] \quad /i = 0 :$$

$$P_0(x) = x^3 - x + 1, \quad 0 \leq x \leq 1.$$

$$\bullet [x_i, x_{i+1}] = [x_1, x_2] = [1, 2] \quad /i = 1 :$$

$$P_1(x) = -x^3 + 6x^2 - 7x + 3, \quad 1 \leq x \leq 2.$$

$$\text{Donc : } P(x) = \begin{cases} x^3 - x + 1, & 0 \leq x \leq 1, \\ -x^3 + 6x^2 - 7x + 3, & 1 \leq x \leq 2. \end{cases}$$

## 2.2 Erreur d'interpolation

Malgré que les valeurs de  $f$  et de son polynôme d'interpolation soient les mêmes aux nœuds d'interpolation. Mais, elles diffèrent en général en tout autre point, donc il est fondamental d'étudier l'**erreur d'interpolation** sur l'intervalle auquel appartiennent les nœuds d'interpolation.

Naturellement, sauf cas particulier, l'expression de l'erreur ne permet pas son calcul exact (car, s'il en était ainsi, il n'y aurait plus d'erreurs) ; elle peut cependant être très utile pour en calculer une majoration.

On peut exprimer l'erreur d'interpolation de la façon suivante :

$$f(x) = P_n(x) + E_n(x) \quad \text{ou encore} \quad E_n(x) = f(x) - P_n(x).$$

On constate immédiatement que l'erreur d'interpolation est nulle aux points de collocation puisque le polynôme passe exactement par ces points. Maintenant, il reste à évaluer cette erreur.

### 2.2.1 Erreur d'interpolation de Lagrange

**Théorème 2.2.1** *On suppose que la fonction  $f$  est de classe  $C^{n+1}([a, b]; \mathbb{R})$  tel que :*  
 $a \leq x_0 < x_1 < \dots < x_n \leq b.$

*Alors,  $\forall x \in [a, b]$ , il existe un point  $\xi \in ]\min(x_0, x), \max(x, x_n)[ \subset [a, b]$  tel que :*

$$E_n(x) := f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^n (x - x_i), \quad (2.9)$$

où

$$\prod_{i=0}^n (x - x_i) = (x - x_0)(x - x_1) \dots (x - x_n).$$

**Preuve.** • Si le point  $x$  coïncide avec l'un des nœuds d'interpolation ( i.e :  $x = x_i$  pour  $i \in \overline{0, n}$ ), les deux membres de (2.9) sont nuls et l'égalité est trivialement vérifiée.

• Supposons maintenant que  $x$  est un point fixé de  $[a; b]$ , tel que :  $x \neq x_i$  pour  $i = \overline{0, n}$ .

Soit  $P_{n+1}(t)$  le polynôme d'interpolation de  $f(t)$  aux points  $x, x_0, \dots, x_{n+1}$ , de sorte que  $P_{n+1} \in \mathcal{P}_{n+1}$ . Par construction  $f(x) - P_n(x) = P_{n+1}(x) - P_n(x)$ . Or le polynôme  $P_{n+1} - P_n$  est de degré  $\leq n + 1$  et s'annule aux  $(n + 1)$  points  $x_0, x_1, \dots, x_n$ . On a donc

$$P_{n+1}(t) - P_n(t) = k \cdot \prod_{i=0}^n (t - x_i), \quad k \in \mathbb{R}.$$

Considérons la fonction

$$g(t) = f(t) - P_{n+1}(t) = f(t) - P_n(t) - k \cdot \prod_{i=0}^n (t - x_i).$$

Cette fonction s'annule en les  $(n + 2)$  points  $x, x_0, \dots, x_n$  donc d'après le lemme (1.1.1) il existe  $\xi \in ]\min(x_0, x), \max(x, x_n)[$  tel que  $g^{(n+1)}(\xi) = 0$ . Or

$$P_n^{(n+1)} = 0, \quad \left( \prod_{i=0}^n (x - x_i) \right)^{(n+1)} = (n + 1)!.$$

On a par conséquent  $g^{(n+1)}(\xi) = f^{(n+1)}(\xi) - k \cdot (n + 1)! = 0$ , d'où

$$f(x) - P_n(x) = P_{n+1}(x) - P_n(x) = k \cdot \prod_{i=0}^n (x - x_i) = \frac{f^{(n+1)}(\xi)}{(n + 1)!} \prod_{i=0}^n (x - x_i).$$

■

**Remarque 2.2.1** *La formule (2.9) ne permet évidemment pas de calculer la valeur exacte de l'erreur parce que, en générale,  $\xi$  est inconnu. Elle permet par contre, d'en calculer une majoration (et permet même de choisir les points  $x_0, x_1, \dots, x_n$  de façon optimale lorsque ceux-ci ne sont pas imposés). D'où :*

**Corollaire 2.2.1** *Sous les mêmes hypothèses du théorème précédent on a :*

$$|f(x) - P_n(x)| \leq \frac{M_{n+1}}{(n + 1)!} \left| \prod_{i=0}^n (x - x_i) \right|,$$

où  $M_{n+1} = \max_{x \in [a,b]} |f^{(n+1)}(x)|$ .

L'erreur d'interpolation résulte de deux termes : le premier terme  $\max_{x \in [a,b]} |f^{(n+1)}(x)|$  dépend de  $f$  et on ne peut pas l'améliorer car  $f$  est donnée, par contre le deuxième terme  $|\prod_{i=0}^n (x - x_i)|$  dépend de la distribution des points  $x_i$ . On peut choisir l'ensemble des points  $\{x_i\}_{i=0, \dots, n}$  pour que l'erreur  $|\prod_{i=0}^n (x - x_i)|$  soit minimal.

**Remarque 2.2.2** *Le polynôme d'interpolation de Newton conduit à la même estimation d'erreur que le polynôme de Lagrange.*

## 2.2.2 Erreur d'interpolation d'Hermite

**Théorème 2.2.2** *On suppose que la fonction  $f$  est de classe  $C^{2n+2}([a, b]; \mathbb{R})$  tel que :  $a \leq x_0 < x_1 < \dots < x_n \leq b$ .*

*Alors,  $\forall x \in [a, b]$ , il existe un point  $\xi \in ]\min(x_0, x), \max(x, x_n)[ \subset [a, b]$  tel que :*

$$E_n(x) := f(x) - P_{2n+1}(x) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \prod_{i=0}^n (x - x_i)^2,$$

*De plus, si  $f$  est de classe  $C^{2n+1}$  alors*

$$|f(x) - P_{2n+1}(x)| \leq \frac{\|f^{(2n+2)}\|_\infty}{(2n+2)!} \left| \prod_{i=0}^n (x - x_i) \right|^2, \text{ pour tout } x \in [a, b].$$

## 2.2.3 Erreur d'interpolation par les fonctions splines

**Par spline linéaire**

**Théorème 2.2.3** *Soit  $[a, b]$  un intervalle contenant  $x_0, x_1, \dots, x_n$ . On suppose que la fonction  $f$  est de classe  $C^2$  sur  $[a, b]$ . Alors, pour tout  $x \in [x_i, x_{i+1}]$ , il existe  $\xi_i \in ]x_i, x_{i+1}[$  tel que :*

$$E_n(x) := f(x) - P_i(x) = \frac{f''(\xi_i)}{2} (x - x_i)(x - x_{i+1}),$$

d'où

$$\forall x \in [x_i, x_{i+1}], \quad |f(x) - P_i(x)| \leq \frac{h_i^2}{8} \max_{t \in [x_i, x_{i+1}]} |f''(t)|,$$

avec  $h = \max_{0 \leq i \leq n-1} h_i$ ,  $tq : h_i = x_{i+1} - x_i$ ,

ce qui conduit à

$$\|f - P_i\|_\infty \leq \frac{h^2}{8} \|f''\|_\infty.$$

Cette estimation montre que l'erreur d'interpolation peut être rendue aussi petite que souhaité, de manière uniforme sur l'intervalle  $[a, b]$ , en prenant  $h$  suffisamment petit.

### Par spline cubique

**Théorème 2.2.4** Soit  $f \in C([a, b])$  avec  $\max_{a \leq x \leq b} |f^{(4)}(x)| = M$ . Si  $P_i(x)$  est l'unique spline cubique serrée interpolant  $f$  aux nœuds  $a = x_0 < x_1 < \dots < x_n = b$ , alors

$$\max_{a \leq x \leq b} |f(x) - P_i(x)| \leq \frac{5M}{384} \max_{0 \leq i \leq n-1} (x_{i+1} - x_i)^4.$$

# Chapitre 3

## Applications avec Matlab

Dans ce chapitre, nous voulons voir des applications très précises sur les différentes méthodes d'interpolation par l'utilisation de logiciel Matlab. Nous pouvons donc comparer ces méthodes et connaître la manière la plus précise et efficace. Pour cela, il est donc préférable de fixer les mêmes points d'interpolation et de les appliquer dans les différentes méthodes d'interpolation afin de pouvoir faire une comparaison exacte.

### Exemple

Dans tout les méthodes d'interpolation on va construire le polynôme d'interpolation  $P(x)$  associée à la fonction  $y = \exp(x)$  aux abscisses  $\{-1, 0, 1\}$ , et dans chaque méthode d'interpolation on va passer aux les quatre étapes suivantes :

1. On va déterminer d'abord ce polynôme de façon analytique.
2. On va écrire un algorithme Matlab permettant l'implémentation de la méthode d'interpolation correspondante.
3. On va tracer, sur la même figure, la fonction et le polynôme d'interpolation.
4. On va faire des discussions.

### 3.1 Application de la méthode de Lagrange

1. D'abord nous déterminerons, analytiquement, le polynôme  $P_2(x)$ .

Et par l'utilisation de la formule d'interpolation de Lagrange (2.1) et pour  $i = 0, 1, 2$ , il vient :

$$\begin{aligned}
 P_2(x) &= \sum_{i=0}^2 y_i L_i(x) \\
 &= y_0 L_0(x) + y_1 L_1(x) + y_2 L_2(x) \\
 &= y_0 \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + y_1 \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + y_2 \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} \\
 &= (\exp)^{-1} \frac{(x-0)(x-1)}{((-1)-0)((-1)-1)} + \frac{(x-(-1))(x-1)}{(0-(-1))(0-1)} + (\exp) \frac{(x-(-1))(x-0)}{(1-(-1))(1-0)} \\
 \implies P_2(x) &= \left( \frac{(\exp)^{-1} - 2 + \exp}{2} \right) x^2 + \left( \frac{\exp - (\exp)^{-1}}{2} \right) x + 1.
 \end{aligned}$$

2. Voici le script Matlab :

```

clear all ; close all ; clc ;

n=3 ; x=[-1 0 1 ] ; y=exp(x) ;

m=10 ; dx=(x(3)-x(1))/m ; xx=x(1) :dx :x(3) ; polyn=0 ; figure('color',[1 1 1]) ;

plot(xx,exp(xx),'-r','LineWidth',2)

for i=1 :n

    lag=1 ;

    for j=1 :n

        if (i~=j)

            lag=((xx-x(j))./(x(i)-x(j))).*lag ;

        end

    end

    polyn=polyn+lag.*y(i) ;

end

```

```

hold on

plot(xx,polyn,'--b','LineWidth',2);

for i=1 :n
    plot(x(i),y(i),'k.','MarkerSize',25,'LineWidth',1.5);hold on
end

xlabel('\fontsize{12}\fontname{Tex} x');
ylabel('\fontsize{12}\fontname{Tex} y=exp(x)');
legend('Solution Exacte','Solution Approchée',2);

```

3. Graphique représentant la fonction et le polynôme d'interpolation :

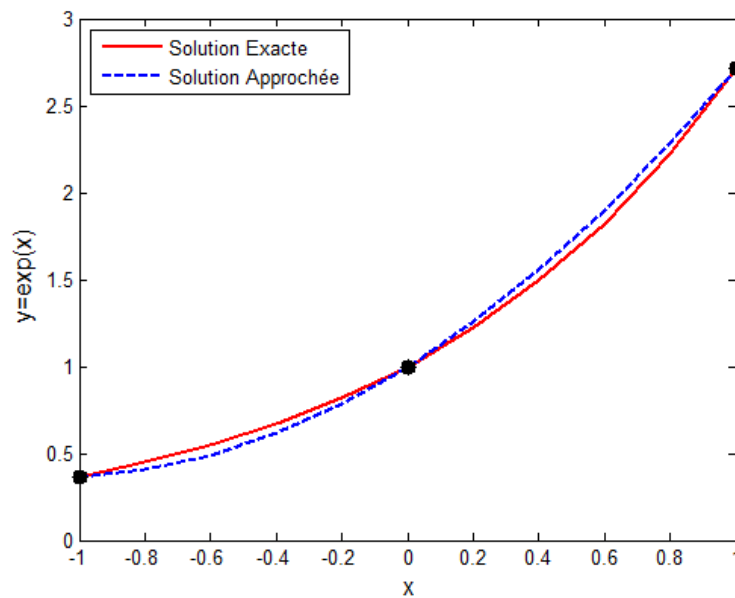


FIG. 3.1 – Interpolation selon Lagrange de  $y = \exp(x)$ .

4. **Discussion** : D'après cette figure, on voit que la solution exacte (trait continu) et la solution numérique (trait discontinu) sont bien rapprochées avec un décalage minimal, ce qui montre la convergence de la méthode de Lagrange pour donner des solutions proches de celles exactes.



## 3.2 Application de la méthode de Newton

1. D'abord nous déterminerons, analytiquement, le polynôme  $P_2(x)$ .

On va construire le tableau des différences divisées d'après (2.2) :

$i$	$x_i$	$f(x_i)$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$
0	-1	$(\exp)^{-1}$		
1	0	1	$1 - (\exp)^{-1}$	$\frac{\exp - 2 + (\exp)^{-1}}{2}$
2	1	$\exp$	$\exp - 1$	

Et par l'utilisation de la formule d'interpolation de Newton (2.3), il vient :

$$\begin{aligned}
 P_2(x) &= f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\
 &= (\exp)^{-1} + (1 - (\exp)^{-1})(x + 1) + \left(\frac{\exp - 2 + (\exp)^{-1}}{2}\right)x(x + 1) \\
 &= \left(\frac{(\exp)^{-1} - 2 + \exp}{2}\right)x^2 + \left(\frac{\exp - (\exp)^{-1}}{2}\right)x + 1.
 \end{aligned}$$

2. Voici le script Matlab :

```

clear all;close all;clc;
n=3;x=[-1 0 1];y=exp(x);
m=10;dx=(x(3)-x(1))/m;xx=x(1):dx:x(3);pol=1;figure('color',[1 1 1]);
plot(xx,exp(xx),'-r','LineWidth',2)
for i=1:n % calcul des differences
    t(i)=y(i);
    for j=i-1:-1:1
        t(j)=(t(j+1)-t(j))./(x(i)-x(j));
    end
    a(i)=t(1);

```

```
end

pol=a(n); % calcul du polynome

for i=n-1 :-1 :1

    pol=pol.*(xx-x(i))+a(i);

end

hold on

plot(xx,pol,'--b','LineWidth',2)

for i=1 :n

    plot(x(i),y(i),'k.','MarkerSize',25,'LineWidth',1.5);hold on

end

xlabel('\fontsize{12}\fontname{Tex} x');

ylabel('\fontsize{12}\fontname{Tex} y=exp(x)');

legend('Solution Exacte','Solution Approchée',2);
```

3. Graphique représentant la fonction et le polynôme d'interpolation :

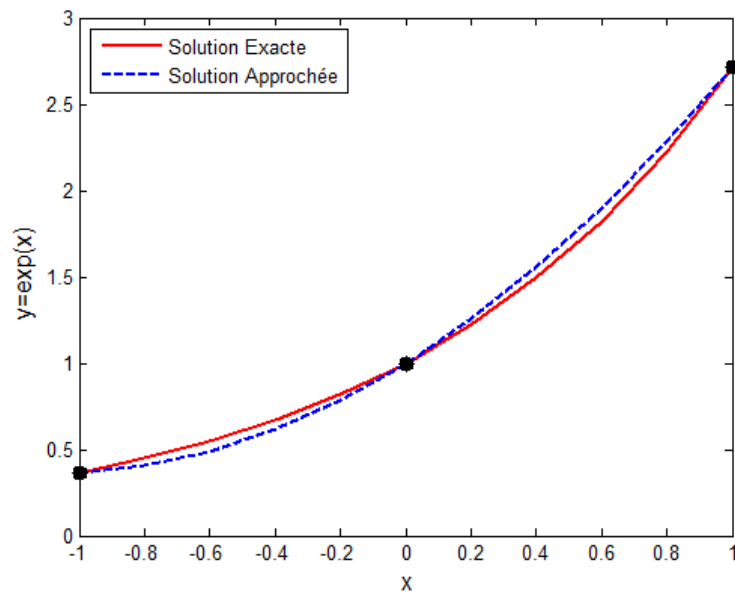


FIG. 3.2 – Interpolation selon Newton de  $y = \exp(x)$ .

4. **Discussion :** Cette figure nous affiche deux courbes (celui la solution exacte et l'autre de la solution numérique donnée par la méthode de Newton) dont on remarque que ces deux courbes sont bien proches l'une de l'autre avec petite marge d'erreur. Telle que dans l'intervalle  $x \in [-0.2; 0.2]$  on observe que les deux courbes sont superposées contrairement en dehors de cet intervalle les deux courbes s'écartent.

### 3.3 Application de la méthode d'Hermite

1. D'abord nous déterminerons, analytiquement, le polynôme  $P_5(x)$ .

Et par l'utilisation de la formule d'interpolation d'Hermite (2.6) et pour  $i = 0, 1, 2$ , il vient :

$$\begin{aligned}
 P_5(x) &= \sum_{i=0}^2 f(x_i)H_i(x) + \sum_{i=0}^2 f'(x_i)V_i(x) \\
 &= \sum_{i=0}^2 f(x_i)[1 - 2(x - x_i)L'_i(x_i)]L_i^2(x) + \sum_{i=0}^2 f'(x_i)(x - x_i)L_i^2(x) \\
 &= f(x_0) \left( 1 - 2(x - x_0) \left( \frac{1}{x_0 - x_1} + \frac{1}{x_0 - x_2} \right) \right) \left( \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} \right)^2 \\
 &\quad + f'(x_0)(x - x_0) \left( \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} \right)^2 + f'(x_1)(x - x_1) \left( \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} \right)^2 \\
 &\quad + f(x_1) \left( 1 - 2(x - x_1) \left( \frac{1}{x_1 - x_0} + \frac{1}{x_1 - x_2} \right) \right) \left( \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} \right)^2 \\
 &\quad + f(x_2) \left( 1 - 2(x - x_2) \left( \frac{1}{x_2 - x_0} + \frac{1}{x_2 - x_1} \right) \right) \left( \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} \right)^2 \\
 &\quad + f'(x_2)(x - x_2) \left( \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} \right)^2 \\
 &= (\exp)^{-1} \left( 1 - 2(x + 1) \left( \frac{-3}{2} \right) \right) \left( \frac{x(x - 1)}{2} \right)^2 + (\exp)^{-1} (x + 1) \left( \frac{x(x - 1)}{2} \right)^2 \\
 &\quad + (x^2 - 1)^2 + x(x^2 - 1)^2 \\
 &\quad + \exp \left( 1 - 2(x - 1) \left( \frac{3}{2} \right) \right) \left( \frac{x(x + 1)}{2} \right)^2 + \exp(x - 1) \left( \frac{x(x + 1)}{2} \right)^2
 \end{aligned}$$

Donc :

$$P_5(x) = \left( (\exp)^{-1} - \left( \frac{\exp}{2} \right) + 1 \right) x^5 + \left( \frac{-3}{4} (\exp)^{-1} - \frac{1}{4} \exp + 1 \right) x^4 \\ + \left( \frac{-3}{2} (\exp)^{-1} + \exp - 2 \right) x^3 + \left( \frac{5}{4} (\exp)^{-1} + \frac{3}{4} \exp - 2 \right) x^2 + x + 1.$$

2. Voici le script Matlab :

```
clear all;close all;clc;
n=3;x=[-1 0 1];y=exp(x);
m=10;dx=(x(3)-x(1))/m;
yy=[y(1),(y(2)-y(1))/(x(2)-x(1)),(y(3)-y(2))/(x(3)-x(2))];
xx = x(1) :dx :x(3); poly = 0;figure('color',[1 1 1]);
plot(xx,exp(xx),'-r','LineWidth',2)

for i=1 :n
    lag=1;ci=0;
    for j=1 :n
        if (i ~= j)
            lag = ((xx - x(j))./(x(i) - x(j))).*lag;
            ci=ci+(1/(x(i) - x(j)));
        end
    end

    Di=1-2.*(xx-x(i)).*ci;
    poly=poly+(y(i).*Di+yy(i).*(xx-x(i))).*(lag).^2;
end

hold on

plot(xx,poly,'--b','LineWidth',2)

for i=1 :n
```

```

    plot(x(i),y(i),'k.','MarkerSize',25,'LineWidth',1.5);hold on
end

xlabel('\fontsize{12}\fontname{Tex} x');
ylabel('\fontsize{12}\fontname{Tex} y=exp(x)');
legend('Solution Exacte','Solution Approchée',2);

```

3. Graphique représentant la fonction et le polynôme d'interpolation :

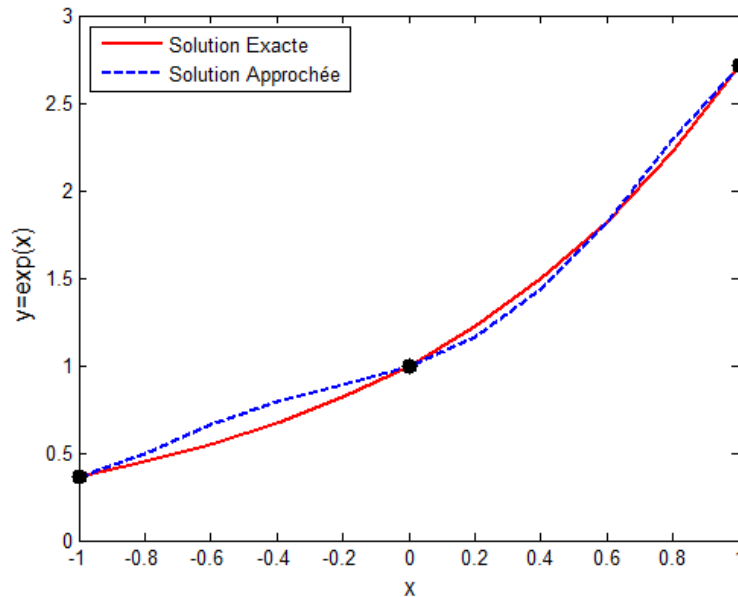


FIG. 3.3 – Interpolation selon Hermite de  $y = \exp(x)$ .

4. **Discussion** : D'après le graphe on remarque que :

- Pour  $x = 0$  et  $x \simeq 0.6$  : Il ya une superposition entre les courbes représentant la solution exacte et la solution numérique (méthode d'Hermite).
- Pour  $x \in [-1, 0[$  : Les deux courbes sont lointe l'une par rapport à l'autre ce qui signifie une erreur remarquable.
- Pour  $x \in ]0, 0.6[$  et  $]0.6, 1]$  : Les deux courbes sont un peu proches l'une de l'autre.

## 3.4 Application de la méthode des fonctions splines

### 3.4.1 Par spline linéaire

1. D'abord nous déterminerons, analytiquement, le polynôme  $P(x)$  de degré 1.

Et par l'utilisation de la formule d'interpolation par spline linéaire (2.7), il vient :

- Sur l'intervalle  $[-1, 0]$ , on obtient :

$$\begin{aligned} P_0(x) &= f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0), \text{ pour } x_0 \leq x \leq x_1 \\ &= f(-1) + \frac{f(0) - f(-1)}{0 + 1}(x + 1), \text{ pour } -1 \leq x \leq 0 \\ &= (1 - (\exp)^{-1})x + 1. \end{aligned}$$

- Sur l'intervalle  $[0, 1]$ , on obtient :

$$\begin{aligned} P_1(x) &= f(x_1) + \frac{f(x_2) - f(x_1)}{x_2 - x_1}(x - x_1), \text{ pour } x_1 \leq x \leq x_2 \\ &= f(0) + \frac{f(1) - f(0)}{1 - 0}(x - 0), \text{ pour } 0 \leq x \leq 1 \\ &= (\exp - 1)x + 1. \end{aligned}$$

Donc :

$$P(x) = \begin{cases} (1 - (\exp)^{-1})x + 1 & \text{pour } -1 \leq x \leq 0 \\ (\exp - 1)x + 1 & \text{pour } 0 \leq x \leq 1 \end{cases}$$

2. Voici le script Matlab :

```
clear all;close all;clc;
fun=@(x) exp(x);n=2;a=-1;b=1;h=1;
x=a :h :b;fx=fun(x);m=10;dx=(b-a)/m;xx=a :dx :b;figure('color',[1 1 1]);
plot(xx,fun(xx),'-r','LineWidth',2);hold on;
for i=1 :n
```

```

step=(x(i+1)-x(i))/n;xi=x(i) :step :x(i+1) ;
polynom=fx(i)+((fx(i+1)-fx(i))/(x(i+1)-x(i)))*(xi-x(i)) ;
plot(xi,polynom,'--b','LineWidth',2);hold on
end

plot(x,fun(x),'k.','MarkerSize',25,'LineWidth',1.5);
xlabel('\fontsize{12}\fontname{Tex} x');
ylabel('\fontsize{12}\fontname{Tex} y=exp(x)');
legend('Solution Exacte','Solution Approchée',2);

```

3. Graphique représentant la fonction et le polynôme d'interpolation :

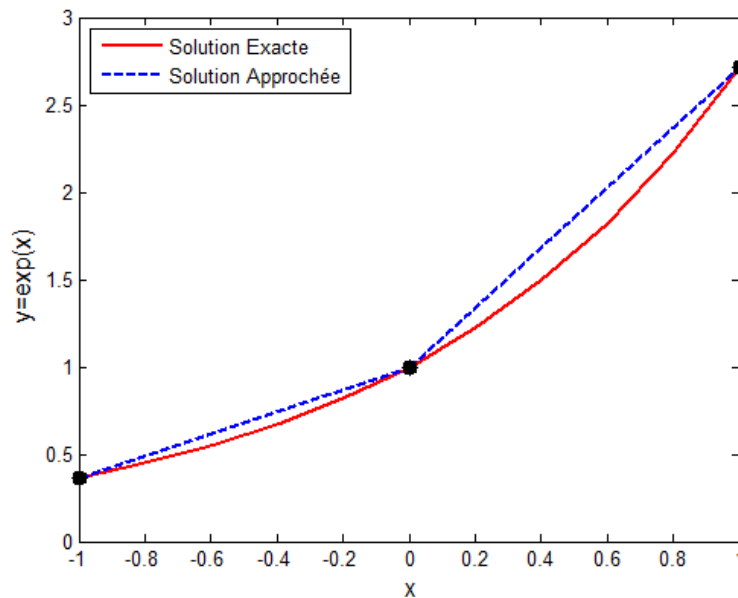


FIG. 3.4 – Interpolation par splines linéaires de  $y = \exp(x)$ .

4. **Discussion :** À travers le graphe obtenu on remarque qu'il y a un décalage entre la représentation exacte (courbe rouge) et la représentation numérique (courbe bleu) surtout sur l'intervalle  $[0, 1]$ , cela explique la faiblesse de l'interpolation par spline linéaire provient du fait que la fonction d'interpolation est continue mais n'est pas dérivable.

### 3.4.2 Par spline cubique

1. D'abord nous déterminerons, analytiquement, le polynôme  $P(x)$  de degré 3.

Et par l'utilisation de la formule d'interpolation par spline cubique (2.8), et pour tout  $x \in [x_i, x_{i+1}]$ , il vient :

$$\begin{aligned} P_i(x) &= a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \\ &= \frac{1}{6h} (\alpha_{i+1} - \alpha_i) (x - x_i)^3 + \frac{1}{2} \alpha_i (x - x_i)^2 + \frac{1}{h} (f(x_{i+1}) - f(x_i)) (x - x_i) \\ &\quad - \frac{h}{6} (2\alpha_i + \alpha_{i+1}) (x - x_i) + f(x_i). \end{aligned}$$

$$\alpha_0 = \alpha_2 = 0, \alpha_1 = ?$$

$$h(\alpha_{i-1} + 4\alpha_i + \alpha_{i+1}) = \frac{6}{h}(f(x_{i+1}) - 2f(x_i) + f(x_{i-1})), \text{ et } \alpha_i = f''(x_i),$$

$$i = 1 : \alpha_0 + 4\alpha_1 + \alpha_2 = 6(f(x_2) - 2f(x_1) + f(x_0)) \Rightarrow 4\alpha_1 = 6(\exp - 2 + (\exp)^{-1})$$

$$\Rightarrow \alpha_1 = \frac{3}{2}(\exp - 2 + (\exp)^{-1}).$$

$$\alpha_0 = 0, \alpha_1 = \frac{3}{2}(\exp - 2 + (\exp)^{-1}), \alpha_2 = 0.$$

- $[x_i, x_{i+1}] = [x_0, x_1] = [-1, 0] \quad /i = 0 :$

$$P_0(x) = \frac{1}{2}(\frac{1}{2} \exp - 1 + \frac{1}{2} (\exp)^{-1})x^3 + \frac{3}{2}(\frac{1}{2} \exp - 1 + \frac{1}{2} (\exp)^{-1})x^2 + \frac{1}{2}(\exp - (\exp)^{-1})x + 1.$$

- $[x_i, x_{i+1}] = [x_1, x_2] = [0, 1] \quad /i = 1 :$

$$P_1(x) = -\frac{1}{4}(\exp - 2 + (\exp)^{-1})x^3 + \frac{3}{4}(\exp - 2 + (\exp)^{-1})x^2 + \frac{1}{2}(\exp - (\exp)^{-1})x + 1.$$

Donc :

$$P(x) = \begin{cases} \text{pour tout } -1 \leq x \leq 0 : \\ \frac{1}{2}(\frac{1}{2} \exp - 1 + \frac{1}{2} (\exp)^{-1})x^3 + \frac{3}{2}(\frac{1}{2} \exp - 1 + \frac{1}{2} (\exp)^{-1})x^2 + \frac{1}{2}(\exp - (\exp)^{-1})x + 1. \\ \text{pour tout } 0 \leq x \leq 1 : \\ -\frac{1}{4}(\exp - 2 + (\exp)^{-1})x^3 + \frac{3}{4}(\exp - 2 + (\exp)^{-1})x^2 + \frac{1}{2}(\exp - (\exp)^{-1})x + 1. \end{cases}$$

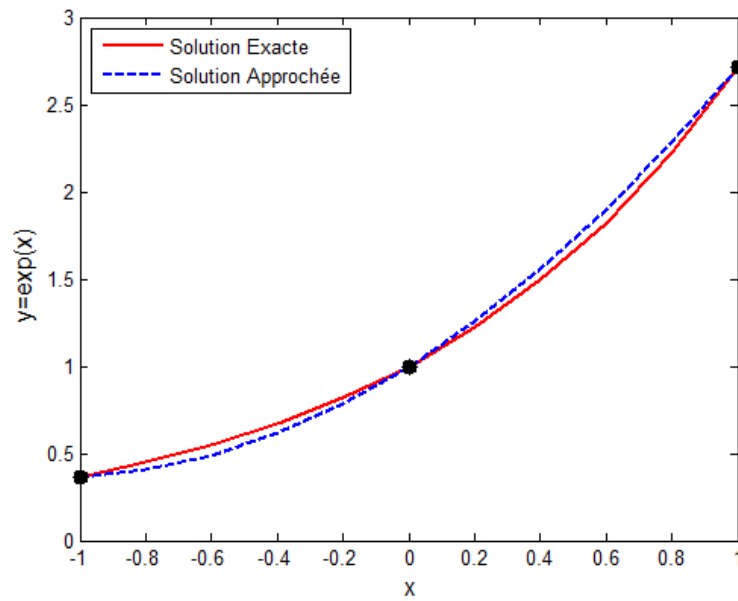
2. Voici le script Matlab :

```
clear all;close all;clc;
```



```
fun=@(x) exp(x);n=2;a=-1;b=1;h=1;
x=a :h :b;fx=fun(x);m=10;dx=(b-a)/m;xx=a :dx :b;
for in=1 :n-1
    bn(in)=(fx(in)-2*fx(in+1)+fx(in+2))*(6/h);
end
sur_diag=diag(ones(n-2,1),1);des_diag=diag(ones(n-2,1),-1);
in_diag=diag(ones(n-1,1));in_diag(in_diag==1)=4;
mtriceA=sur_diag+des_diag+in_diag;alpha=bn*inv(mtriceA)';
alpha=[0,alpha,0];
figure('color',[1 1 1]);
plot(xx,fun(xx),'-r','LineWidth',2);
hold on;
for i=1 :n
    ai=(alpha(i+1)-alpha(i))/(6*h);bi=alpha(i)/2;
    ci=(fx(i+1)-fx(i))/h-(2*alpha(i)+alpha(i+1))*(h/6);
    step=(x(i+1)-x(i))/n;
    xi=x(i) :step :x(i+1);
    polynom=ai*(xi-x(i)).^3+bi*(xi-x(i)).^2+ci*(xi-x(i))+fx(i);
    plot(xi,polynom,'--b','LineWidth',2);hold on;
end
plot(x,fun(x),'k.','MarkerSize',25,'LineWidth',1.5);
xlabel('\fontsize{12}\fontname{Tex} x');
ylabel('\fontsize{12}\fontname{Tex} y=exp(x)');
legend('Solution Exacte','Solution Approchée',2);
```

3. Graphique représentant la fonction et le polynôme d'interpolation :

FIG. 3.5 – Interpolation par splines cubique de  $y = \exp(x)$ .

4. **Discussion :** À partir de cette figure, on peut dire que la solution exacte (trait continue) et la solution numérique (trait discontinue) sont bien proches l'une de l'autre avec une petite marge d'erreur, ce qui montre la convergence de la méthode de spline cubique pour donner des solutions proches à celles exactes.

# Conclusion

Nous continuons ce que nous avons abordé dans le troisième chapitre de nous représentons les différentes méthodes d'interpolation en utilisant logiciel Matlab, mais cette fois dans la même courbe afin de connaître la méthode la plus efficace et la plus précise. Nous avons donc obtenu cette figure :

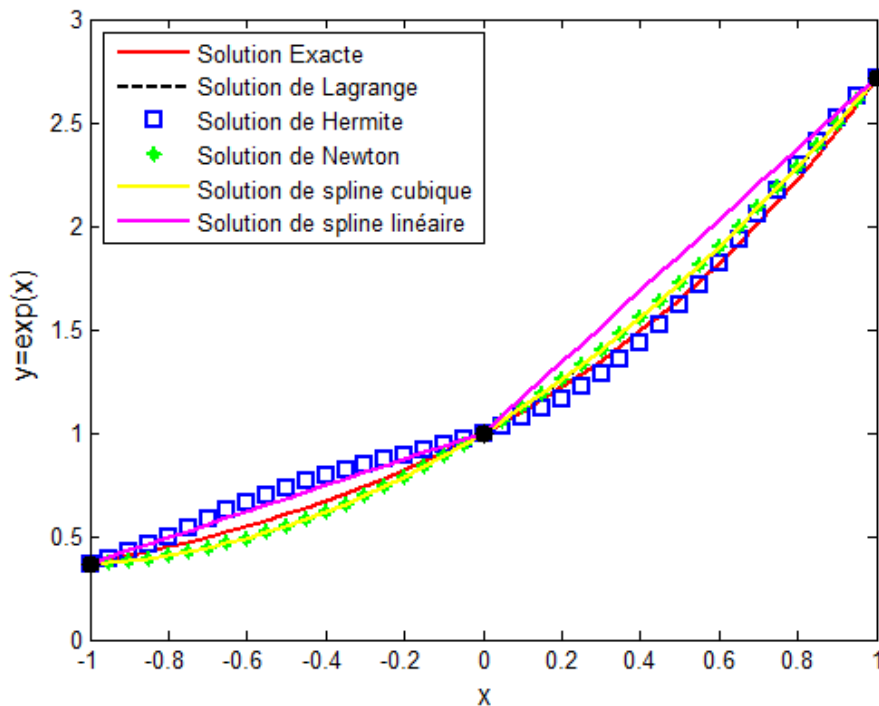


FIG. 3.6 – Comparaison entre la solution exacte ( $y = \exp(x)$ ) et leurs solutions approchées.

**Discussion :** D'après cette figure on remarque que :

- Les courbes des méthodes d'interpolation Lagrange, Newton et Spline cubique coïncident

les uns des autres et leur proximité avec la solution exacte.

- La courbe de la méthode d'Hermite est quelque peu différente de la solution exacte.
- La méthode de Spline linéaire est loin de son prédécesseur.

Nous pouvons donc conclure que les meilleures méthodes d'interpolation sont les méthodes de Lagrange, Newton et Spline cubique. Être le plus proche de la solution exacte (la fonction d'origine).

À la fin de notre travail on a constaté que les méthodes d'interpolation facilitent la manipulation et ramener à des calculs simples en utilisant quelques des polynômes.

# Bibliographie

- [1] Ariba, Y. Cadieux, J. Manuel Matlab. Icam de Toulouse. Version 0.1.
- [2] Azerad, P. Cuer, M. (13 Décembre 2013). Licence de mathématiques : Troisième année GLMA 506 Analyse Numérique des équations différentielles ordinaires. Université de Montpellier II.
- [3] Cluzeau, T. Analyse numérique. École nationale supérieure d'ingénieurs de Limoges. [http://www.unilim.fr/pages\\_perso/thomas.cluzeau](http://www.unilim.fr/pages_perso/thomas.cluzeau).
- [4] Demailly, J. P. (2006). Analyse numérique et équations différentielles. Nouvelle édition. EDP sciences.
- [5] Faccanoni, G. (5 Juin 2014). Analyse numérique recueil d'exercices corrigés et aide-mémoire. <http://faccanoni.univ-tln.fr/enseignements.html>.
- [6] Fischer, P. Weynans, L. Dossal, C. Approximation numérique. Université Bordeaux 1, France. [Patrick.Fischer@math.u-bordeaux1.fr](mailto:Patrick.Fischer@math.u-bordeaux1.fr).
- [7] Fortin, A. Analyse numérique pour ingénieurs. Editions de l'école Polytechnique de Montréal.
- [8] Grivet, J. P. (2009). Méthodes numériques appliquées pour le scientifique et l'ingénieur. EDP sciences.
- [9] Guilpin, C. (2012). Manuel de calcul numérique appliqué. EDP sciences.
- [10] Jedrzejewski, F. (2005). Introduction aux méthodes numériques. Deuxième édition.
- [11] Kenouche, S. Méthodes numériques et programmation. Université Mohamed Khider de Biskra. <http://sites.univ-biskra.dz/kenouche>.

- [12] Lakrib, M. (2007). Cours d'analyse numérique troisième édition. Université Djillali Liabes Sidi Bel Abbès.
- [13] Leborgne, G. (5 Juin 2012). Interpolation polynomiale, intégration numérique, résolution numérique d'équations différentielles.
- [14] Legendre, G. (2009/2010). Méthodes numériques introduction à l'analyse numérique et au calcul scientifique. Université Dauphine Paris.
- [15] Legendre, G. (29 Janvier 2018). Méthodes numériques introduction à l'analyse numérique et au calcul scientifique. Université Dauphine Paris.
- [16] Mebarki, K. Analyse numérique cours, 2ème année licence mathématiques. Université Abderrahmane Mira de Béjaia.
- [17] Postel, M. (septembre 2004). Introduction au logiciel Matlab. Université Pierre et Marie Curie.
- [18] Pujo-Menjouet, L. Introduction à l'analyse numérique deuxième année de licence. Université Claude Bernard, Lyon I.
- [19] Saad, M. (2011/2012). Analyse numérique. Ecole Centrale de Nantes. Mazen Saad@ec-nantes.fr.

# Annexe : Logiciel Matlab

**MATLAB** est une abréviation de Matrix LABoratory, il est un logiciel de calcul numérique commercialisé par la société MathWorks. Il a été initialement développé à la fin des années 70 par Cleve Moler, professeur de mathématique à l'université du Nouveau-Mexique puis à Stanford, pour permettre aux étudiants de travailler à partir d'un outil de programmation de haut niveau et sans apprendre le Fortran ou le C.

MATLAB est un logiciel interactif basé sur le calcul matriciel (MATrix LABoratory). Il est utilisé dans les calculs scientifiques et les problèmes d'ingénierie parce qu'il permet de résoudre des problèmes numériques complexes en moins de temps requis par les langages de programmation, et ce grâce à une multitude de fonctions intégrées et à plusieurs programmes outils testés et regroupés selon usage (boîtes à outils ou Toolbox).

Le logiciel propose un véritable environnement de travail composé de multiples fenêtres. Nous pouvons distinguer quatre blocs :

- **Command window** (console d'exécution) : à l'invite de commande « >> », l'utilisateur peut entrer les instructions à exécuter. Il s'agit de la fenêtre principale de l'interface.
- **Current directory** (répertoire courant) : permet de naviguer et de visualiser le contenu du répertoire courant de l'utilisateur. Les programmes de l'utilisateur doivent être situés dans ce répertoire pour être visible et donc exécutable.
- **Workspace** (espace de travail) : permet de visualiser les variables définies, leur type, la taille occupée en mémoire...

- **Command history** : historique des commandes que l'utilisateur a exécutées. Il est possible de faire glisser ces commandes vers la fenêtre de commande.

On peut regrouper quelques commandes et fonctions de Matlab dans le tableau suivant :

Fonction	Description
clear	Efface variables et fonctions de l'espace de travail
clc	Efface la fenêtre des commandes
all	Vrai si tous les éléments d'un vecteur sont non nuls
for	Répétition
if	Instruction conditionnelle
end	Terminaison de if, for et while
xlabel	Légende axe horizontal
ylabel	Légende axe vertical
figure	Ouvre une nouvelle fenêtre graphique
plot( $x, y$ )	Tracé de la courbe passant par les points ( $x, y$ )
legend('lab1','lab2','lab3',...)	Légende avec une chaîne de caractères pour chaque courbe
hold on / off	Pour superposer ou non des courbes
inv( $A$ )	Inverse d'une matrice
diag	Crée ou extrait une matrice diagonale
ones( $i, j$ )	Crée un tableau de $i$ lignes $j$ colonnes contenant des 1
~ =	Différent de
%	Commentaires