

UNIVERSITÉ DE LA POLYNÉSIE FRANÇAISE

---

---

PROGRAMMATION  
ANALYSE NUMÉRIQUE

---

LICENCE 2 - MATHS INFO

---

---

BENJAMIN GRAILLE  
Université Paris-Saclay

15 avril 2022



# Table des matières

<b>1</b>	<b>Introduction et généralités</b>	<b>5</b>
1	Nécessité de l'analyse numérique . . . . .	6
2	Rappels d'analyse . . . . .	6
2.1	Régularité . . . . .	7
2.2	Formules de Taylor . . . . .	8
2.3	Convexité . . . . .	8
<b>2</b>	<b>Interpolation polynomiale</b>	<b>11</b>
1	Interpolation . . . . .	11
2	Exemples avec peu de points . . . . .	13
2.1	Interpolation à un seul point . . . . .	13
2.2	Interpolation à deux points : exemple de la droite . . . . .	13
2.3	Interpolation à trois points . . . . .	15
3	Polynôme interpolateur de Lagrange . . . . .	17
3.1	Définition et propriétés . . . . .	17
3.2	Expression dans la base canonique . . . . .	18
3.3	Formule de Lagrange et poids barycentriques . . . . .	20
3.4	Comportement asymptotique lorsque $n$ tend vers l'infini . . . . .	23
<b>3</b>	<b>Intégration numérique</b>	<b>27</b>
1	Formules de quadratures . . . . .	27
1.1	Définitions et premières propriétés . . . . .	28
1.2	Formules de quadratures élémentaires et composées . . . . .	31
2	Méthodes de quadrature classiques . . . . .	32
2.1	Une liste des méthodes classiques . . . . .	33
2.2	Analyse des méthodes des rectangles . . . . .	35
2.3	Analyse de la méthode des trapèzes . . . . .	37
2.4	Analyse de la méthode du point milieu . . . . .	38
2.5	Analyse de la méthode de Simpson . . . . .	40
2.6	Contrôle de l'erreur . . . . .	42
3	Les algorithmes des méthodes de quadrature classiques . . . . .	43
<b>4</b>	<b>Résolution d'équations ordinaires</b>	<b>45</b>
1	Exemples d'application . . . . .	45
1.1	Schémas numériques pour équations différentielles ordinaires . . . . .	45
1.2	Méthode de tir pour les problèmes aux limites du second ordre . . . . .	47
2	Position correcte du problème . . . . .	47
2.1	Existence de solution . . . . .	48
2.2	Notion de conditionnement . . . . .	48
2.3	vitesse de convergence - ordre de convergence . . . . .	50
3	Méthodes de type encadrement . . . . .	51
3.1	méthode de la dichotomie . . . . .	51
3.2	Méthode de la sécante . . . . .	53
4	Méthodes de type interpolation . . . . .	56

4.1	méthode de Newton . . . . .	57
4.2	méthode de la fausse position . . . . .	60

<b>Bibliographie</b>		<b>63</b>
----------------------	--	-----------

# 1 Introduction et généralités

Dans ce cours, nous décrivons les premiers outils de l'analyse numérique et nous proposons une implémentation en `python` des différents modèles numériques rencontrés.

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

Plus formellement, l'analyse numérique est l'étude des algorithmes permettant de résoudre numériquement par discrétisation les problèmes de mathématiques continues (distinguées des mathématiques discrètes). Cela signifie qu'elle s'occupe principalement de répondre de façon numérique à des questions à variable réelle ou complexe comme l'algèbre linéaire numérique sur les champs réels ou complexes, la recherche de solution numérique d'équations différentielles et d'autres problèmes liés survenant dans les sciences physiques et l'ingénierie. Branche des mathématiques appliquées, son développement est étroitement lié à celui des outils informatiques.

D'une manière assez caricaturale, nous pouvons définir l'analyse numérique comme l'ensemble des mathématiques appliquées dont l'objet d'étude est centré sur des algorithmes informatiques. Elle est constituée de trois blocs inter-dépendants :

**modélisation** ce bloc consiste à proposer un modèle mathématique qui devra représenter le plus fidèlement possible le phénomène réel qui nous intéresse ;

**calcul scientifique** ce bloc consiste à proposer et à implémenter un algorithme permettant de calculer une solution approchée du modèle mathématique ;

**analyse** ce bloc consiste à analyser mathématiquement les propriétés du modèle et de "sa" (ou "ses") solution(s).

Afin de mener à bien l'ensemble de ce programme, de nombreuses branches de mathématiques appliquées ou d'analyse numérique ont vu le jour. Nous pouvons citer par exemple les équations différentielles ordinaires (ODEs), les équations aux dérivées partielles (PDEs), l'optimisation, l'analyse du signal, l'algèbre linéaire...

Dans ce chapitre, pour mieux comprendre ce qu'est l'analyse numérique, nous nous intéressons à un exemple de problème de dynamique de population, c'est-à-dire que nous essayons de déterminer un ou plusieurs modèles permettant de décrire l'évolution d'une population d'individus au cours du temps. Nous ne prenons pas en compte les effets spatiaux pour simplifier : une seule variable est utilisée pour décrire l'évolution, il s'agit du temps. En particulier, cela signifie que nos individus ne peuvent pas se déplacer dans l'espace (afin de se regrouper ou de se répartir sur un territoire par exemple).

L'objectif est évidemment de décrire le ou les modèles mais également d'étudier leurs solutions. L'écriture du problème mathématique est intéressante en elle-même mais sa ou ses solutions le sont encore plus pour bien des applications. Nous serons alors amenés à introduire et utiliser de nombreux concepts et outils mathématiques pour étudier les solutions d'un point de vue théorique (existence, unicité, propriétés de régularité, de périodicité,...) mais aussi d'un point de vue numérique. En effet, pour de nombreux modèles, il est impossible (c'est un théorème) d'écrire de manière simple la solution à l'aide de sommes, produits, composées de fonctions, éventuellement fonctions mathématiques classiques ( $\cos$ ,  $\sin$ ,  $\exp$ ,  $\ln$ ,...). Afin de réellement prédire l'évolution de la population qui nous intéresse, ne restera alors que la simulation numérique, c'est-à-dire le calcul approché par un ordinateur de la solution exacte.

## 1 NÉCESSITÉ DE L'ANALYSE NUMÉRIQUE

Supposons que nous souhaitons résoudre un problème général similaire aux modèles continus de la dynamique des populations :

$$\begin{cases} u'(t) = f(t, u(t)), & t > 0, \\ u(0) = u_0, \end{cases} \quad (1.1)$$

où  $u_0 \in \mathbb{R}$  et  $f : (t, x) \mapsto f(t, x) \in \mathcal{C}^1(\mathbb{R}_+^* \times \mathbb{R})$ .

Il n'est en général pas possible de donner une expression analytique de la solution  $t \mapsto u(t)$  (cette solution est bien définie de manière unique par le théorème de Cauchy-Lipschitz). Il peut alors être intéressant de déterminer un algorithme permettant de trouver une solution approchée du problème de Cauchy 1.1. Une méthode pour y parvenir est d'intégrer l'équation entre  $t = 0$  et  $t > 0$  :

$$u(t) = u_0 + \int_0^t u'(s) \, ds = u_0 + \int_0^t f(s, u(s)) \, ds.$$

Nous sommes alors ramener à déterminer une valeur approchée d'une intégrale.

De même, il n'existe pas nécessairement de primitive analytique de la fonction  $s \mapsto f(s, u(s))$ , surtout que la fonction  $u$  n'est pas connue... Pour construire des méthodes de quadrature, c'est-à-dire des méthodes numériques d'approximation d'intégrales, il est possible de remplacer la fonction à intégrer par un polynôme qui "ressemble" à la fonction à intégrer.

L'objectif de ce cours est de construire les outils d'analyse numérique permettant de trouver des méthodes numériques (et de les analyser) approchant la solution du problème 1.1. Nous définissons un pas de temps  $\Delta t > 0$  et nous construisons une suite  $(u^n)_{n \in \mathbb{N}}$  telle que  $u^n$  approche  $u(t^n)$  avec  $t^n = n\Delta t$ . L'idée est de remplacer la fonction  $f$  par un polynôme  $P_n$  et de poser

$$u^{n+1} = u^n + \int_{t^n}^{t^{n+1}} P_n(s) \, ds.$$

Nous pouvons alors récrire le problème sous la forme

$$u^{n+1} = u^n + \phi(u^n, u^{n+1}),$$

où  $\phi$  est une fonction régulière. Nous devons donc peut-être résoudre une équation pour déterminer  $u^{n+1}$  en fonction de  $u^n$ .

Nous devons donc

- ▷ proposer une méthode pour construire le polynôme  $P_n$  à partir de quelques valeurs ( $u^n$  et éventuellement  $u^{n+1}$ ) : nous étudierons **la théorie de l'interpolation polynomiale** ;
- ▷ proposer une méthode de quadrature pour approcher l'intégrale par celle d'un polynôme interpolateur : nous étudierons **la théorie de l'intégration numérique** ;
- ▷ proposer des méthodes pour résoudre des équations ordinaires pour calculer  $u^{n+1}$  en fonction de  $u^n$  : nous étudierons **les méthodes de résolution approchée d'équations ordinaires**.

## 2 RAPPELS D'ANALYSE

Nous souhaitons à présent rappeler quelques résultats essentiels en analyse numérique autour des formules de Taylor. En effet, l'analyse locale des fonctions passe le plus souvent par un

développement asymptotique et les formules de Taylor sont très utiles dès que les fonctions sont suffisamment régulières.

Tous les résultats sont présentés sans démonstration car ce sont des prérequis de ce cours.

## 2.1 RÉGULARITÉ

Nous commençons par rappeler quelques définitions concernant la régularité des fonctions réelles à valeurs réelles, ainsi que des théorèmes fondamentaux en analyse.

### DÉFINITION 1.1 – continuité

Soit  $f : I \rightarrow \mathbb{R}$  où  $I$  est un intervalle de  $\mathbb{R}$ . Soit  $x \in I$ . La fonction  $f$  est dite *continue* en  $x$  si

$$\forall \varepsilon > 0 \exists \eta > 0 : y \in I \cap ]x-\eta, x+\eta[ \implies |f(x) - f(y)| \leq \varepsilon.$$

De plus, la fonction  $f$  est dite *continue sur  $I$*  si elle est continue en tout point  $x$  de  $I$ .

### THÉORÈME 1.2 – Valeurs intermédiaires

Soit  $f : I \rightarrow \mathbb{R}$  une fonction continue. Soit  $(a, b) \in I^2$  tel que  $f(a) < 0 < f(b)$ . Alors il existe  $c \in ]a, b[$  tel que  $f(c) = 0$ .

Ce théorème fondamental en analyse peut aussi se résumer par la phrase : « l'image d'un segment par une fonction continue est aussi un segment ». La démonstration peut se faire de manière constructive à l'aide de l'algorithme de la dichotomie.

### DÉFINITION 1.3 – dérivabilité

Soit  $f : I \rightarrow \mathbb{R}$  où  $I$  est un intervalle de  $\mathbb{R}$ . La fonction  $f$  est dite *dérivable* en  $x$  si la limite

$$\lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

existe et est finie. Dans ce cas, nous appellerons *dérivée* en  $x$  cette limite et nous la noterons  $f'(x)$ . De plus, la fonction  $f$  est dite *dérivable sur  $I$*  si elle est dérivable en tout point  $x$  de  $I$ .

### THÉORÈME 1.4 – Rolle

Soit  $f : [a, b] \rightarrow \mathbb{R}$  une fonction continue sur  $[a, b]$  et dérivable sur  $]a, b[$ . Supposons que  $f(a) = f(b)$ . Alors il existe  $c \in ]a, b[$  tel que  $f'(c) = 0$ .

### THÉORÈME 1.5 – Accroissements finis

Soit  $f : [a, b] \rightarrow \mathbb{R}$  une fonction continue sur  $[a, b]$  et dérivable sur  $]a, b[$ . Alors il existe  $c \in ]a, b[$  tel que

$$f(b) - f(a) = f'(c)(b - a).$$

**DÉFINITION 1.6**

Soit  $I$  un intervalle ouvert de  $\mathbb{R}$ . Nous définissons par récurrence les espaces  $\mathcal{C}^k(I)$ ,  $k \in \mathbb{N}$ ,

$$\begin{aligned}\mathcal{C}^0(I) &= \left\{ f : I \rightarrow \mathbb{R} \text{ continue sur } I \right\}, \\ \mathcal{C}^k(I) &= \left\{ f : I \rightarrow \mathbb{R} \text{ dérivable sur } I \text{ telle que } f' \in \mathcal{C}^{k-1}(I) \right\}, \quad k > 0.\end{aligned}$$

**2.2 FORMULES DE TAYLOR**

Les formules de Taylor sont toujours démontrées à partir du théorème des accroissements finis et sont basées sur la remarque évidente suivante

$$f(x+h) - f(x) = h \int_0^1 f'(x+ht) dt.$$

Tous les théorèmes proposés ont la même base mais les hypothèses sont légèrement différentes (sur la régularité de la fonction) et il est en général utile de réfléchir au choix de la “bonne” formule selon ce que l’on veut démontrer.

**THÉORÈME 1.7 – Taylor avec reste intégral**

Soit  $f : I \rightarrow \mathbb{R} \in \mathcal{C}^{n+1}(I)$ . Alors pour tout  $x \in I$  et  $h \in \mathbb{R}$  tel que  $x+h \in I$

$$f(x+h) = \sum_{k=0}^n \frac{h^k}{k!} f^{(k)}(x) + h^{n+1} \int_0^1 \frac{1}{n!} f^{(n+1)}(x+hs)(1-s)^n ds. \quad (1.2)$$

**THÉORÈME 1.8 – Taylor – Lagrange**

Soit  $f : I \rightarrow \mathbb{R} \in \mathcal{C}^n(I)$  dérivable jusqu’à l’ordre  $n+1$ . Alors pour tout  $x \in I$  et  $h \in \mathbb{R}$  tel que  $x+h \in I$  il existe  $\xi \in ]0, 1[$  tel que

$$f(x+h) = \sum_{k=0}^n \frac{h^k}{k!} f^{(k)}(x) + \frac{h^{n+1}}{(n+1)!} f^{(n+1)}(x+h\xi). \quad (1.3)$$

**THÉORÈME 1.9 – Taylor – Young**

Soit  $f : I \rightarrow \mathbb{R} \in \mathcal{C}^{n-1}(I)$  dérivable jusqu’à l’ordre  $n$ . Alors pour tout  $x \in I$  et  $h \in \mathbb{R}$  tel que  $x+h \in I$

$$f(x+h) = \sum_{k=0}^n \frac{h^k}{k!} f^{(k)}(x) + R_n(h), \quad (1.4)$$

où la fonction  $R_n$  est négligeable devant la fonction  $h \mapsto h^n$  au voisinage de 0. C’est-à-dire

$$\lim_{h \rightarrow 0, h \neq 0} \frac{R_n(h)}{h^n} = 0.$$

**2.3 CONVEXITÉ**

Nous rappelons les définitions d’une fonction convexe, strictement convexe, concave et strictement concave. Ces fonctions sont très importantes en analyse et au moins localement pour les



fonctions régulières, il s'agit de cas générique.

**DÉFINITION 1.10 – fonction convexe**

Soit  $f : [a, b] \rightarrow \mathbb{R}$ . La fonction  $f$  sera dite *convexe* si

$$\forall (x, y) \in [a, b]^2, x \neq y, \quad \forall \lambda \in ]0, 1[, \quad f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y).$$

La fonction sera dite *strictement convexe* si l'inégalité est stricte.

Finalement une fonction  $f$  sera dite *concave* si  $-f$  est convexe.

**PROPOSITION 1.11 – convexité pour les fonctions régulières**

Soit  $f : [a, b] \rightarrow \mathbb{R}$  de classe  $\mathcal{C}^2$ . La fonction  $f$  est convexe si, et seulement si,  $f''$  est positive.

*Démonstration.* Nous allons prouver plus exactement l'équivalence entre être convexe et avoir une dérivée croissante pour une fonction  $f$  de classe  $\mathcal{C}^1$ .

Supposons que  $f$  est convexe. Soit  $x < y$ . Nous comparons  $f'(x)$  et  $f'(y)$  en utilisant la corde entre ces deux points. Nous pouvons récrire la convexité en prenant

$$\lambda x + (1 - \lambda)y = x + h \implies (1 - \lambda)(y - x) = h \implies \lambda = 1 - \frac{h}{y - x}.$$

Nous en déduisons que

$$\frac{f(x + h) - f(x)}{h} \leq \frac{f(y) - f(x)}{y - x}.$$

Nous récrivons ensuite la convexité en prenant

$$\lambda x + (1 - \lambda)y = y - h \implies \lambda(y - x) = h \implies \lambda = \frac{h}{y - x}.$$

Nous en déduisons que

$$\frac{f(y) - f(x)}{y - x} \leq \frac{f(y) - f(y - h)}{h}.$$

En passant à la limite  $h \rightarrow 0$ , nous obtenons que la fonction  $f'$  est croissante, donc  $f''$  est positive.

Réciproquement, si  $f''$  est positive, la fonction  $f'$  est croissante. Soit  $(x, y) \in [a, b]^2$  et soit  $\lambda \in [0, 1]$ . Pour fixer les idées, supposons que  $x \leq y$ . Posons  $z = \lambda x + (1 - \lambda)y$ . Nous avons donc  $x \leq z \leq y$ .

$$f(y) - f(z) = \int_z^y f'(t) dt = \lambda(y - x) \int_0^1 f'(sy + (1 - s)z) ds,$$

$$f(y) - f(x) = \int_x^y f'(t) dt = (y - x) \int_0^1 f'(sy + (1 - s)x) ds.$$

Comme  $f'$  est croissante,

$$\forall s \in [0, 1] \quad sy + (1 - s)z \geq sy + (1 - s)x \implies f'(sy + (1 - s)z) \geq f'(sy + (1 - s)x).$$

Nous concluons

$$f(y) - f(z) \geq \lambda f(y) - \lambda f(x) \iff f(z) \leq \lambda f(x) + (1 - \lambda)f(y),$$

ce qui termine la preuve en reprenant l'expression de  $z = \lambda x + (1 - \lambda)y$ . ◻

Nous pouvons également remarquer que si une fonction est de classe  $\mathcal{C}^2$  avec une dérivée seconde strictement positive, alors la fonction est strictement convexe. La réciproque n'est pas vraie car une fonction strictement convexe peut avoir une dérivée seconde qui s'annule (par exemple en un point comme la fonction  $x \mapsto x^4$ ). Cependant, il est possible de démontrer que la dérivée

seconde d'une fonction strictement convexe ne peut pas s'annuler sur un intervalle de longueur non nulle.

## 2 Interpolation polynomiale

En analyse numérique, les polynômes de Lagrange, du nom de Joseph-Louis Lagrange, permettent d'interpoler une série de points par un polynôme qui passe exactement par ces points appelés aussi nœuds. Cette technique d'interpolation polynomiale a été découverte par Edward Waring en 1779 et redécouverte plus tard par Leonhard Euler en 1783.

Edward Waring (1736–1798) était un mathématicien britannique. Il est entré au Magdalene College de Cambridge en tant que sizar, et est devenu Senior wrangler en 1757. Waring a écrit un certain nombre d'articles dans les *Philosophical Transactions of the Royal Society*, traitant de la résolution d'équations algébriques, de théorie des nombres, de séries, de l'approximation des racines, de l'interpolation, de la géométrie des sections coniques et de la dynamique.

Leonhard Euler (1707–1783) est un mathématicien et physicien suisse, qui passa la plus grande partie de sa vie dans l'Empire russe et en Allemagne. Il était notamment membre de l'Académie royale des sciences de Prusse à Berlin. Euler fit d'importantes découvertes dans des domaines aussi variés que le calcul infinitésimal et la théorie des graphes. Il introduisit également une grande partie de la terminologie et de la notation des mathématiques modernes, en particulier pour l'analyse mathématique, comme la notion de fonction mathématique. Il est aussi connu pour ses travaux en mécanique, en dynamique des fluides, en optique et en astronomie ou en géométrie du triangle.

Joseph Louis de Lagrange (1736–1813) né de parents français descendants de Descartes, est un mathématicien, mécanicien et astronome sarde, naturalisé français. En mathématiques, il fonde le calcul des variations, avec Euler, et la théorie des formes quadratiques. En physique, en précisant le principe de moindre action, avec le calcul des variations, vers 1756, il invente la fonction de Lagrange, qui vérifie les équations de Lagrange, puis développe la mécanique analytique, vers 1788, pour laquelle il introduit les multiplicateurs de Lagrange. Il entreprend aussi des recherches importantes sur le problème des trois corps en astronomie, un de ses résultats étant la mise en évidence des points de libration (dits points de Lagrange) (1772).



Joseph Louis de Lagrange



Edward Waring



Leonhard Euler

---

### 1 INTERPOLATION

---

L'interpolation d'une fonction de  $\mathbb{R}$  dans  $\mathbb{R}$  est un problème ancien qui a de très nombreuses applications que ce soit d'un point de vue purement théorique comme l'intégration numérique (et la résolution d'équations différentielles par conséquent), la construction d'algorithmes de résolution de  $f(x) = 0$  ou d'un point de vue industriel comme la visualisation de données, la représentation 3D d'objet. Essentiellement, l'objectif est de remplacer la fonction compliquée (ou le nuage de points) qui nous est donnée par une fonction beaucoup plus simple : un polynôme. Malheureusement deux concepts différents apparaissent rapidement : l'interpolation et l'approximation.

Si nous imaginons un nuage de points (correspondant par exemple à un ensemble discret de valeurs  $(x_i, f(x_i))$  pour une certaine fonction  $f : \mathbb{R} \rightarrow \mathbb{R}$ ), il est possible :

- ▷ de l'interpoler, c'est-à-dire de faire passer un polynôme par tous les points ;
- ▷ de l'approcher (par exemple au sens des moindres carrés), c'est-à-dire de trouver un polynôme de bas degré dont la différence pour une certaine norme est petite.

A la figure Fig. 2.1, nous trouvons des illustrations de l'interpolation et de l'approximation par

un polynôme de degré 2 lorsque le nombre de points  $N$  varie. La fonction  $f$  associée à ces nuages de points est une perturbation aléatoire de la fonction  $x \rightarrow 4x(1-x)$ .

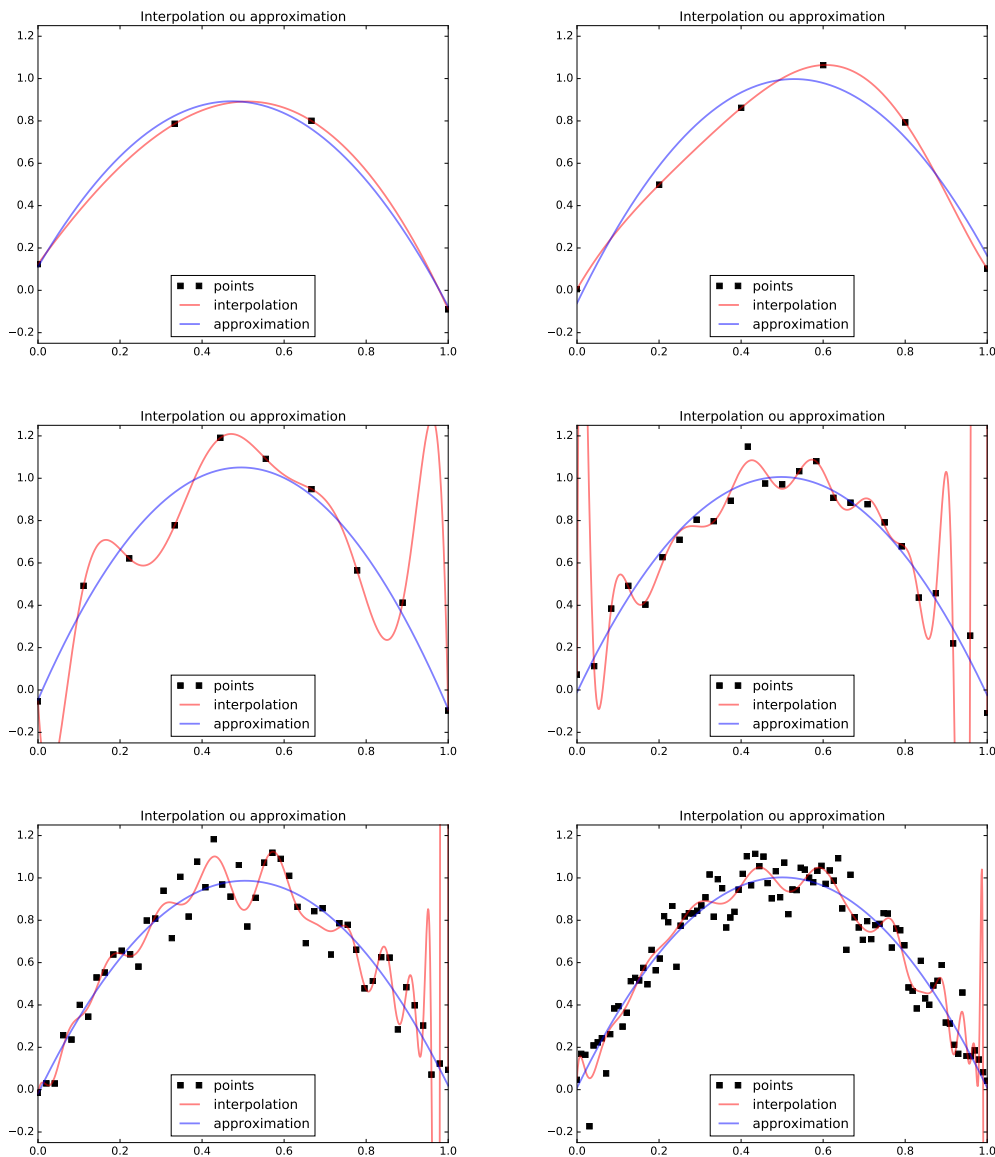


FIGURE 2.1 – Interpolation et approximation pour  $N \in \{4, 6, 10, 25, 50, 100\}$ .

Nous pouvons remarquer (sans savoir pour le moment comment l'interpolée a été calculée) que le polynôme interpolateur passe par les bons points lorsque  $N$  est petit (moins que 10 dans notre exemple) mais qu'il est mal calculé lorsque  $N$  devient grand : bien que par définition il devrait passer par tous les points d'interpolation, le calcul retourne un polynôme qui ne passe pas par ces points. Ainsi, il faut bien comprendre qu'interpoler ne signifie pas approcher et que lorsque le nombre de points est élevé, l'interpolation perd de son intérêt. Cependant, cette notion d'interpolation est très utile en analyse numérique et en calcul scientifique pour approcher (oui cette fois c'est vrai) localement une fonction à l'aide de quelques points d'interpolation. Nous nous en servons pour proposer des méthodes de calcul d'intégrales approchées, des formules de résolution d'équations différentielles, des méthodes de résolution d'équations non linéaires...

Dans ce chapitre, nous nous intéresserons essentiellement à l'interpolation par des polynômes interpolateurs de Lagrange, mais nous verrons également que ce n'est pas la seule. Nous commençons par nous intéresser à quelques exemples simples avant de généraliser les résultats obtenus.

## 2 EXEMPLES AVEC PEU DE POINTS

Pour bien comprendre la notion de polynôme interpolateur, nous allons commencer par des exemples pour  $N$  petit, où  $N$  est le nombre de points par lequel nous imposons que le polynôme doit passer.

### 2.1 INTERPOLATION À UN SEUL POINT

Prenons dans cette partie  $N = 1$ , c'est-à-dire que nous cherchons à faire passer un polynôme par un point de coordonnées  $(x_1, y_1)$ . Il est évident qu'il en existe une infinité si le degré du polynôme n'est pas limité. Nous pouvons donc chercher un ou le polynôme de degré le plus petit possible passant par ce point. Une illustration est donnée à la figure 2.2.

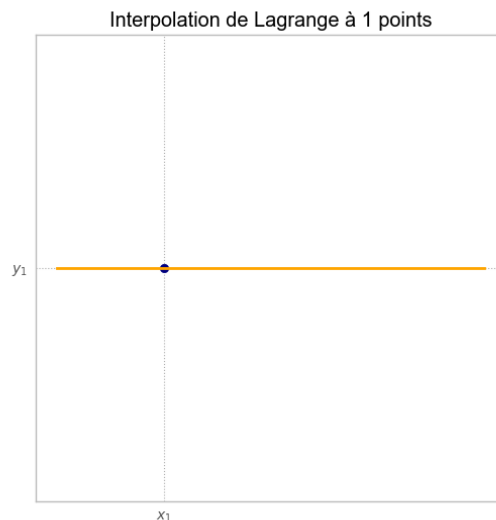


FIGURE 2.2 – Exemple de polynôme interpolateur de Lagrange lorsque  $N = 1$ .

Dans ce cas, nous cherchons le polynôme de degré au plus 0 qui passe par le point de coordonnées  $(x_1, y_1)$ . C'est donc le polynôme constant  $P = y_1$ . Il est évidemment unique.

### 2.2 INTERPOLATION À DEUX POINTS : EXEMPLE DE LA DROITE

Nous nous intéressons à présent à l'exemple le plus simple dont les applications sont malgré tout innombrables : par deux points passe une unique droite. Ce résultat ou plutôt cette définition selon Archimède doit évidemment être suivi d'un calcul analytique afin de déterminer l'équation de cette droite. Dans cette section, nous prenons donc  $N = 2$ .

## 2.2.1 UN PEU DE GÉOMÉTRIE EUCLIDIENNE

**THÉORÈME 2.1 – Polynôme interpolateur de degré 1**

Soit  $(x_1, y_1)$  et  $(x_2, y_2)$  deux points distincts de  $\mathbb{R}^2$ . Il existe une unique droite  $\mathcal{D}$  passant par ces deux points <sup>a</sup> :

$$(x, y) \in \mathcal{D} \iff (x - x_1)(y_2 - y_1) - (y - y_1)(x_2 - x_1) = 0.$$

Si de plus  $x_1 \neq x_2$ , il existe un unique polynôme  $P \in \mathbb{R}_1[X]$  tel que

$$\mathcal{D} = \{(x, P(x)), x \in \mathbb{R}\} \quad \text{avec} \quad P = \frac{(X - x_1)y_2 - (X - x_2)y_1}{x_2 - x_1}.$$

<sup>a</sup>. On distinguera bien  $\mathcal{D}$  qui est un objet géométrique c'est-à-dire une partie du plan, de  $P$  le polynôme (ou parfois fonction polynomiale) dont le graphe est égal à  $\mathcal{D}$ .

*Démonstration.* Nommons les points afin de faire de la géométrie euclidienne! Notons  $M = (x, y)$ ,  $M_1 = (x_1, y_1)$ ,  $M_2 = (x_2, y_2)$ , les trois points de  $\mathbb{R}^2$  qui nous intéressent. Le point  $M$  est sur la droite  $\mathcal{D} = (M_1M_2)$  si, et seulement si, les vecteurs  $\overrightarrow{M_1M}$  et  $\overrightarrow{M_1M_2}$  doivent être colinéaires, ce qui peut se récrire sous la forme

$$\begin{vmatrix} x - x_1 & x_2 - x_1 \\ y - y_1 & y_2 - y_1 \end{vmatrix} = 0.$$

Le calcul de ce déterminant donne directement l'expression de la droite  $\mathcal{D}$ . Puis, dans le cas  $x_1 \neq x_2$ , il est possible de paramétrer cette droite par l'abscisse  $x$  en divisant par  $x_2 - x_1$ . Cela induit l'expression du polynôme  $P$ . ◻

Remarquez l'expression proposée pour le polynôme interpolateur  $P$  qui est symétrique par rapport aux deux points. Nous pouvons en proposer deux autres : la première est la décomposition de ce polynôme dans la base canonique  $(1, X)$

$$P = X \frac{y_2 - y_1}{x_2 - x_1} + \frac{x_2 y_1 - x_1 y_2}{x_2 - x_1},$$

la seconde dans une base "adaptée" aux points d'interpolation  $(1, X - x_1)$

$$P = (X - x_1) \frac{y_2}{x_2 - x_1} + y_1.$$

La question que nous devons nous poser à présent est la suivante : comment calculer le polynôme  $P$  (qui est seulement de degré 1 dans notre exemple) et surtout comment calculer les valeurs de  $P(x)$  pour de nombreuses valeurs de  $x$ . En effet, il est parfois nécessaire de connaître l'expression exacte du polynôme (de chacun de ses coefficients dans une certaine base), parfois plutôt de calculer l'évaluation de ce polynôme un grand nombre de fois.

La réponse à cette question ne peut pas être l'expression trouvée dans ce théorème puisque la démonstration géométrique pour la trouver ne fonctionnera plus lorsque le nombre de points d'interpolation sera supérieur à 2.

## 2.2.2 UNE PREMIÈRE MÉTHODE DE CALCUL

La première méthode qui vient à l'esprit est de décomposer le polynôme recherché dans la base canonique et de résoudre les équations associées. Nous cherchons donc à déterminer deux réels

$a_0$  et  $a_1$ , ces deux réels étant les coefficients dans la base canonique du polynôme  $P$ , c'est-à-dire tels que  $P = a_0 + a_1X$ . Nous écrivons ensuite les relations qui doivent être satisfaites :

$$\begin{cases} a_0 + a_1x_1 = y_1 \\ a_0 + a_1x_2 = y_2 \end{cases} \iff \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}.$$

Comme  $x_1 \neq x_2$ , la matrice est inversible et nous pouvons laisser le soin à `Python` et son package `numpy.linalg` de résoudre le problème : les coefficients  $a_0$  et  $a_1$  existent donc bien et sont uniques.

Allons un peu plus loin dans l'analyse de cette méthode. Nous posons

$$A = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \end{pmatrix}.$$

Nous avons donc

$$A^{-1} = \frac{1}{x_2 - x_1} \begin{pmatrix} x_2 & -x_1 \\ -1 & 1 \end{pmatrix}.$$

Ainsi, si  $x_1$  et  $x_2$  sont proches, le calcul de la matrice  $A^{-1}$  et plus précisément la résolution du système linéaire conduira à des erreurs d'arrondis. D'une manière générale, ce phénomène s'amplifiera lorsque le nombre de points d'interpolation augmente : c'est ce que nous voyons à la Fig. 2.1 où le polynôme interpolateur "numérique" ne passe plus par les points d'interpolation lorsque le nombre de points est trop grand.

---

### 2.2.3 EXPRESSION DU POLYNÔME DANS UNE BASE ADAPTÉE

---

Nous proposons une seconde méthode pour évaluer le polynôme interpolateur. L'idée est de le représenter dans une autre base que la base canonique : une base adaptée aux points d'interpolation. Nous cherchons donc à déterminer deux réels  $a_0$  et  $a_1$  tels que  $P = a_0 + a_1(X - x_1)$ . Le calcul se fait alors plus simplement :

$$P(x_1) = y_1 \implies a_0 = y_1, \quad P(x_2) = y_2 \implies a_1 = \frac{y_2 - y_1}{x_2 - x_1}.$$

Par ailleurs, nous remarquons que les calculs du polynôme interpolateur passant par les points  $(x_1, y_1)$  et  $(x_2, y_2)$ , noté  $P_{1,2}$  ici, peuvent être réutilisés si l'on souhaite ajouter un point d'interpolation.

---

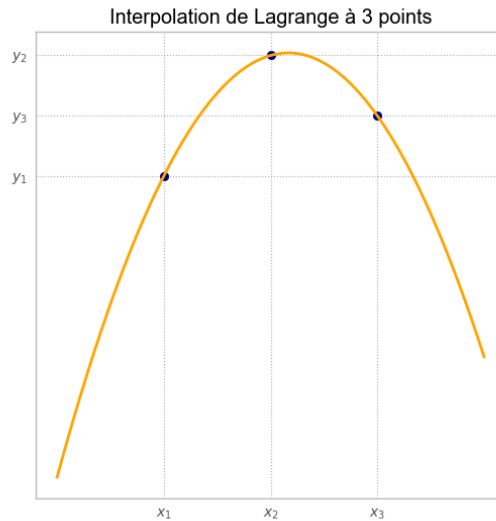
## 2.3 INTERPOLATION À TROIS POINTS

---

Prenons dans cette partie  $N = 3$ . Une illustration est donnée à la figure 2.3.

Dans ce cas, nous cherchons le polynôme de degré au plus 2 qui passe par les trois points de coordonnées  $(x_1, y_1)$ ,  $(x_2, y_2)$  et  $(x_3, y_3)$ . C'est donc une parabole que l'on peut chercher sous la forme  $P = aX^2 + bX + c$ . Les calculs commencent à être plus compliqués et il est nécessaire d'avoir une stratégie pour trouver les coefficients  $a$ ,  $b$  et  $c$ . Ces trois coefficients sont solutions du système linéaire suivant

$$\begin{cases} P(x_1) = ax_1^2 + bx_1 + c = y_1, \\ P(x_2) = ax_2^2 + bx_2 + c = y_2, \\ P(x_3) = ax_3^2 + bx_3 + c = y_3. \end{cases}$$


 FIGURE 2.3 – Exemple de polynôme interpolateur de Lagrange lorsque  $N = 3$ .

Un calcul un peu fastidieux permet de déterminer la valeur des trois coefficients  $a$ ,  $b$  et  $c$  :

$$a = -\frac{y_1(x_2 - x_3) + y_2(x_3 - x_1) + y_3(x_1 - x_2)}{(x_2 - x_3)(x_3 - x_1)(x_1 - x_2)},$$

$$b = \frac{y_1(x_2^2 - x_3^2) + y_2(x_3^2 - x_1^2) + y_3(x_1^2 - x_2^2)}{(x_2 - x_3)(x_3 - x_1)(x_1 - x_2)},$$

$$c = -\frac{y_1x_2x_3(x_2 - x_3) + y_2x_3x_1(x_3 - x_1) + y_3x_1x_2(x_1 - x_2)}{(x_2 - x_3)(x_3 - x_1)(x_1 - x_2)}.$$

Plusieurs remarques peuvent alors être faites :

- ▷ les calculs deviennent rapidement lourds et il faut trouver une méthode générale ;
- ▷ il semble que les coefficients obtenus pour  $N$  points ont un dénominateur  $\prod_{i < j} (x_i - x_j)$  (le même que celui des polynômes de Lagrange).

#### EXERCICE 2.2 – Polynôme interpolateur de degré 2

Déterminez l'expression du polynôme interpolateur passant par les points  $(x_1, y_1)$ ,  $(x_2, y_2)$  et  $(x_3, y_3)$  dans la base adaptée  $(1, (X - x_1), (X - x_1)(X - x_2))$ .

*Solution.* Nous écrivons  $P = a_0 + a_1(X - x_1) + a_2(X - x_1)(X - x_2)$ . Nous avons en évaluant en  $x_1$

$$P(x_1) = a_0 + a_1(x_1 - x_1) + a_2(x_1 - x_1)(x_1 - x_2) = y_1 \quad \implies \quad a_0 = y_1.$$

Puis en évaluant en  $x_2$  :

$$P(x_2) = y_1 + a_1(x_2 - x_1) + a_2(x_2 - x_1)(x_2 - x_2) = y_2 \quad \implies \quad a_1 = \frac{y_2 - y_1}{x_2 - x_1}.$$

Enfin en évaluant en  $x_3$  :

$$P(x_3) = y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x_3 - x_1) + a_2(x_3 - x_1)(x_3 - x_2) = y_3$$

$\implies$

$$a_2 = \frac{y_3 - y_1 - \frac{y_2 - y_1}{x_2 - x_1}(x_3 - x_1)}{(x_3 - x_1)(x_3 - x_2)} = \frac{\frac{y_3 - y_1}{x_3 - x_1} - \frac{y_2 - y_1}{x_2 - x_1}}{x_3 - x_2}$$



### 3 POLYNÔME INTERPOLATEUR DE LAGRANGE

Dans cette section, nous étudions les propriétés du polynôme interpolateur de Lagrange et nous décrivons quelques algorithmes permettant de le calculer.

#### 3.1 DÉFINITION ET PROPRIÉTÉS

Etant donnés  $N$  points du plan  $(x_1, y_1), \dots, (x_N, y_N)$ , avec  $N \in \mathbb{N}^*$ , nous souhaitons construire un polynôme de degré minimal qui passe par tous ces points.

Le premier résultat fondamental est que le degré minimal est  $N-1$ , qu'il est toujours possible de construire un tel polynôme et qu'enfin ce polynôme est unique.

##### THÉORÈME 2.3 – Isomorphisme d'espaces vectoriels

Soit  $(x_1, \dots, x_N) \in \mathbb{R}^N$  tels que  $x_i \neq x_j$  si  $i \neq j$ . L'application

$$\Phi : \begin{cases} \mathbb{R}_{N-1}[X] & \longrightarrow \mathbb{R}^N \\ P & \longmapsto (P(x_1), \dots, P(x_N)) \end{cases}$$

est bijective (c'est un isomorphisme).

*Démonstration.* Pour montrer que l'application linéaire  $\Phi : \mathbb{R}_{N-1}[X] \rightarrow \mathbb{R}^N$  définie par  $\Phi(P) = (P(x_1), \dots, P(x_N))$  est un isomorphisme d'espaces vectoriels, il suffit de vérifier qu'elle est injective, puisque les espaces sont de même dimension  $N$ . Supposons donc que  $P \in \mathbb{R}_{N-1}[X]$  tel que  $\Phi(P) = 0$ . Nous avons donc un polynôme de degré au plus  $N-1$  qui a  $N$  racines, c'est le polynôme nul.  $\bullet$

##### DÉFINITION 2.4 – Polynôme interpolateur de Lagrange

Etant donnés des réels  $x_1, \dots, x_N$  deux à deux distincts et des réels quelconques  $y_1, \dots, y_N$ , l'unique polynôme  $P$  qui vérifie

$$P \in \mathbb{R}_{N-1}[X] \quad \text{et} \quad P(x_k) = y_k \quad \text{pour} \quad 1 \leq k \leq N$$

est appelé le *polynôme interpolateur de Lagrange aux points*  $(x_1, y_1), \dots, (x_N, y_N)$ .

Un cas particulièrement intéressant lorsque l'on fait de l'analyse est le cas où le nuage de points  $(x_1, y_1), \dots, (x_N, y_N)$  est sur le graphe d'une fonction régulière  $f$ . C'est-à-dire que  $f(x_i) = y_i$ ,  $1 \leq i \leq N$ . Dans ce cas, on dira que le polynôme  $P$  est le polynôme interpolateur de la fonction  $f$  aux points  $x_1, \dots, x_N$ .

##### DÉFINITION 2.5 – Polynôme interpolateur de Lagrange

Si  $f$  est une fonction continue sur un intervalle  $[a, b]$  et à valeurs dans  $\mathbb{R}$  et si  $x_1, \dots, x_N$  sont  $N$  points distincts de  $[a, b]$ , alors l'unique polynôme  $P \in \mathbb{R}_{N-1}[X]$  tel que  $P(x_i) = f(x_i)$  pour  $1 \leq i \leq N$ , est appelé polynôme d'interpolation de Lagrange de  $f$  aux points  $x_1, \dots, x_N$ .

Il est possible d'estimer l'écart entre une fonction  $f$  et son polynôme interpolateur de Lagrange lorsque la fonction est régulière. Cette estimation repose sur le lemme suivant.

**LEMME 2.6 – Rolle généralisé**

Soient  $a, b$  deux réels tels que  $a < b$  et  $g : [a, b] \rightarrow \mathbb{R}$  une fonction continue. Si  $g$  s'annule en  $m$  ( $m \geq 2$ ) points  $t_1, \dots, t_m$  deux à deux distincts de  $]a, b[$  et si  $g$  est  $m - 1$  fois dérivable sur  $]a, b[$ , alors il existe  $t \in ]t_1, t_m[$  tel que  $g^{(m-1)}(t) = 0$ .

*Démonstration.* La preuve se fait par récurrence sur  $m \geq 2$ . Pour  $m = 2$ , c'est exactement le théorème de Rolle (Th. 1.4). Supposons le résultat du lemme vrai pour  $m \geq 2$  et prenons  $g$  une fonction  $m$  fois dérivable qui s'annule en  $t_1, \dots, t_{m+1}$  points distincts. D'après le théorème de Rolle 1.4, la fonction  $g'$  s'annule sur chaque intervalle  $]t_i, t_{i+1}[$ ,  $1 \leq i \leq m$ , en un point noté  $\tilde{t}_i$ . La fonction  $g'$  est donc  $m-1$  fois dérivable et s'annule  $m$  fois, par hypothèse de récurrence, il existe  $t \in ]\tilde{t}_1, \tilde{t}_m[ \subset ]t_1, t_{m+1}[$  tel que  $g^{(m)}(t) = 0$ .  $\bullet$

**THÉORÈME 2.7 – Théorème de l'erreur**

Soient  $a, b$  deux réels tels que  $a < b$  et  $f : [a, b] \rightarrow \mathbb{R}$  une fonction continue. Etant donné  $N$  points  $x_1, x_2, \dots, x_N$  deux à deux distincts de  $]a, b[$ , on pose

$$\omega_N(X) = (X - x_1) \dots (X - x_N)$$

et on note  $P \in \mathbb{R}_{N-1}[X]$  le polynôme d'interpolation de Lagrange de  $f$  aux points  $x_1, \dots, x_N$ .

Si  $f$  est  $N$  fois dérivable sur  $]a, b[$ , alors pour tout  $x \in [a, b]$  il existe  $t \in ]a, b[$  tel que

$$f(x) - P(x) = \omega_N(x) \frac{f^{(N)}(t)}{N!}.$$

*Démonstration.* Pour  $x$  fixé dans  $[a, b] \setminus \{x_1, \dots, x_N\}$ , on applique le lemme de Rolle généralisé (Lm. 2.6) à la fonction auxiliaire

$$F(t) = f(t) - P(t) - \frac{f(x) - P(x)}{\omega_N(x)} \omega_N(t).$$

La fonction  $F$  est  $N$  fois dérivable et s'annule aux  $N+1$  points  $x, x_1, \dots, x_N$ . D'après le lemme de Rolle généralisé (Lm. 2.6), il existe  $t \in ]a, b[$  tel que  $F^{(N)}(t) = 0$ . Calculons la dérivée  $N^{\text{ième}}$  de  $F$ .

$$F^{(N)}(t) = f^{(N)}(t) - 0 - \frac{f(x) - P(x)}{\omega_N(x)} N! \implies f(x) - P(x) = \omega_N(x) \frac{f^{(N)}(t)}{N!}.$$

Par ailleurs, si  $x \in \{x_1, \dots, x_N\}$ , nous avons  $f(x) - P(x) = 0$  et  $\omega_N(x) = 0$ , ce qui termine la preuve.  $\bullet$

**COROLLAIRE 2.8**

Si  $f^{(N)}$  est uniformément bornée par  $M$  sur  $]a, b[$ , alors pour tout  $x \in [a, b]$

$$|f(x) - P(x)| \leq \frac{M}{N!} |\omega_N(x)| \leq \frac{M}{N!} (b - a)^N.$$

### 3.2 EXPRESSION DANS LA BASE CANONIQUE

Soit  $N \in \mathbb{N}$ ,  $N \geq 1$ . Etant donné  $N$  réels deux à deux distincts  $x_1, \dots, x_N$  et  $N$  réels quelconques  $y_1, \dots, y_N$ , on note  $P$  le polynôme d'interpolation de Lagrange aux points  $(x_1, y_1), \dots, (x_N, y_N)$ . On souhaite évaluer  $P(x)$  pour  $x \in \mathbb{R} \setminus \{x_1, \dots, x_N\}$  quelconque.

La première idée pour évaluer le polynôme interpolateur est de résoudre un système linéaire satisfait par les coordonnées de ce polynôme dans la base canonique. Cette méthode conduit à

résoudre un système de Vandermonde. Si nous notons

$$P = \sum_{i=0}^{N-1} a_i X^i,$$

l'expression du polynôme interpolateur dans la base canonique, nous avons

$$P(x_k) = \sum_{i=0}^{N-1} a_i x_k^i = y_k$$

ce qui se réécrit sous forme matricielle  $\mathbf{V}(x_1, \dots, x_N) \mathbf{a} = \mathbf{y}$  avec

$$\mathbf{V}(x_1, \dots, x_N) = \begin{pmatrix} x_1^0 & \dots & x_1^{N-1} \\ \vdots & & \vdots \\ x_N^0 & \dots & x_N^{N-1} \end{pmatrix}, \quad \mathbf{a} = \begin{pmatrix} a_0 \\ \vdots \\ a_{N-1} \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}.$$

Cette méthode a deux défauts : la matrice est pleine (donc la résolution du système linéaire est coûteuse) et elle est mal conditionnée (le conditionnement croît comme l'exponentielle de  $N$ ). Nous envisagerons donc d'autres méthodes de calcul. Cependant, lorsque l'expression du polynôme interpolateur est donnée dans la base canonique, il est bon d'avoir un algorithme rapide d'évaluation de ce polynôme en un point  $x$ . L'algorithme de Horner permet de n'effectuer que  $N$  additions et  $N$  multiplications pour réaliser ce calcul.

#### ALGORITHME 2.1 – Horner

Si  $x$  est un réel et si  $Q$  est le polynôme défini par

$$Q(X) = a_0 X^n + a_1 X^{n-1} + \dots + a_{n-1} X + a_n,$$

la suite finie  $q_0, q_1, \dots, q_n$  définie par  $q_0 = a_0$  et  $q_k = q_{k-1}x + a_k$  pour  $k = 1, \dots, n$  vérifie  $q_n = Q(x)$ .

```
def Horner(p, x):
    y = 0
    for ak in p:
        y *= x
        y += ak
    return y
```

*Démonstration.* Il suffit d'écrire  $Q(X) = a_n + X \left[ a_{n-1} + X \left( a_{n-2} + X \left[ \dots \left( a_1 + X(a_0) \right) \right] \right) \right]$ . ◻

La classe `poly1d` du package `numpy` de `python` est une classe pour les polynômes d'une variable. Un polynôme est stocké sous la forme d'un tableau `numpy` contenant les coefficients dans la base canonique dans l'ordre décroissant. Il est possible d'évaluer ce polynôme en un point (ou même en plusieurs points directement) en utilisant l'algorithme de Horner. Par exemple

```
import numpy as np
p = np.poly1d([1, 2, 3]) # polynomial X^2 + 2X + 3
p(2) # p(2) = 11
```

Ainsi, nous pouvons proposer un premier algorithme de calcul de l'interpolation aux points  $(x_i, y_i)$ ,  $1 \leq i \leq N$ .

ALGORITHME 2.2 – *InterpVdM*

```

import numpy as np
def InterpVdM(x, y, xx):
    """
    Interpolation with the Vandermonde method
    """
    x = np.asanyarray(x, dtype = 'float')
    y = np.asanyarray(y, dtype = 'float')
    xx = np.asanyarray(xx, dtype = 'float')
    n = x.size
    if y.size != n:
        print('Error in InterpVdM: x and y do not have the same size')
    A = np.ones((n, n))
    for k in range(1,n):
        A[:, k] = A[:, k-1] * x
    p = np.linalg.solve(A, y)
    yy = Horner(p[:-1], xx)
    return yy if xx.size != 1 else np.asscalar(yy)
    
```

La fonction `InterpVdM` prend trois arguments de type `numpy array` : `x`, `y` et `xx` et calcule le polynôme interpolateur aux points  $(x_i, y_i)_{1 \leq i \leq N}$  puis l'évalue aux points `xx` en utilisant l'algorithme de Horner.

## 3.3 FORMULE DE LAGRANGE ET POIDS BARYCENTRIQUES

Etant donnés  $N$  réels  $(x_1, \dots, x_N)$  deux à deux distincts, l'application

$$\Phi : \begin{cases} \mathbb{R}_{N-1}[X] & \longrightarrow \mathbb{R}^N \\ P & \longmapsto (P(x_1), \dots, P(x_N)) \end{cases}$$

est un isomorphisme d'espaces vectoriels. Nous pouvons alors définir l'image de la base canonique de  $\mathbb{R}^N$  par l'application  $\Phi^{-1}$  et nous la noterons  $(L_1, \dots, L_N)$ . A la figure 2.4, vous trouverez les premiers polynômes de Lagrange pour des points équi-répartis.

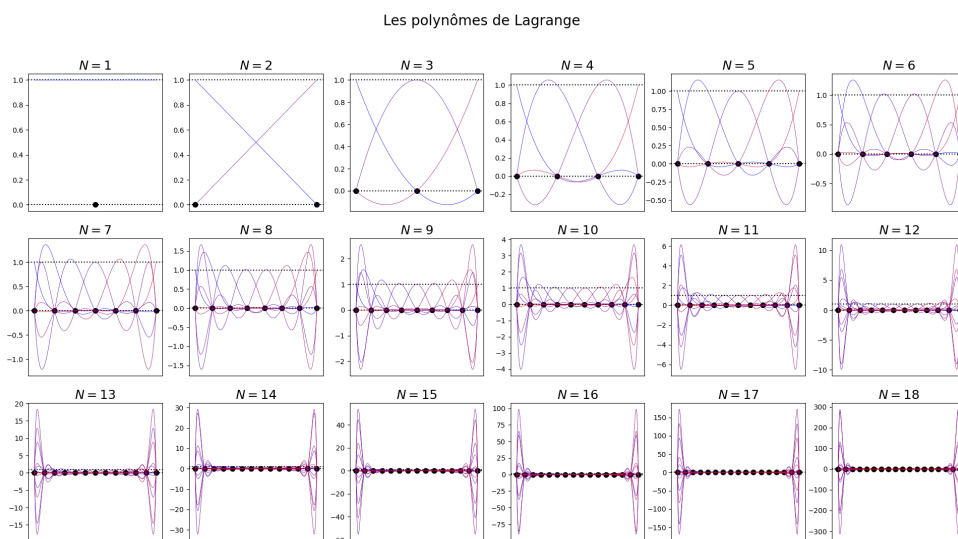


FIGURE 2.4 – Polynôme de Lagrange pour des points équi-répartis.

## DÉFINITION 2.9 – Polynômes de Lagrange

Le  $i$ -ème polynôme de Lagrange, noté  $L_i \in \mathbb{R}_{N-1}[X]$ , pour  $1 \leq i \leq N$  est défini par

$$L_i = \prod_{j \neq i} \frac{X - x_j}{x_i - x_j}, \quad 1 \leq i \leq N.$$

Il est tel que

$$L_i(x_j) = \begin{cases} 1 & \text{si } i = j, \\ 0 & \text{sinon.} \end{cases}$$

En utilisant cette base adaptée aux points  $(x_1, \dots, x_n)$ , nous pouvons donner une nouvelle expression du polynôme interpolateur de Lagrange. Nous avons de manière évidente

$$P(X) = \sum_{i=1}^n y_i L_i(X).$$

Une réécriture astucieuse de cette dernière expression conduit à la formule de Lagrange, très efficace pour évaluer ce polynôme. A nouveau on pose

$$\omega_N(X) = (X - x_1) \dots (X - x_N).$$

## PROPOSITION 2.10 – Formule de Lagrange

On a une formule dite *formule de Lagrange* pour  $P(x)$

$$P(x) = \sum_{i=1}^N y_i \frac{\omega_N(x)}{(x - x_i)\omega'_N(x_i)} \quad \text{ou encore} \quad P(x) = \frac{\sum_{i=1}^N \frac{y_i}{(x - x_i)\omega'_N(x_i)}}{\sum_{i=1}^N \frac{1}{(x - x_i)\omega'_N(x_i)}}$$

valables lorsque  $x$  est un réel distinct de  $x_1, \dots, x_N$ .

*Démonstration.* Les polynômes  $L_1, \dots, L_N$  vérifient

$$L_i(x_i) = 1 \quad \text{et} \quad L_i(x_j) = 0 \quad \text{pour tout } j \in \{1, \dots, N\} \setminus \{i\}.$$

Ces polynômes forment une base de  $\mathbb{R}_{N-1}[X]$  et vérifient  $\sum_{i=1}^N L_i(X) = 1$ .

Pour  $x \in \mathbb{R} \setminus \{x_1, \dots, x_N\}$ , les valeurs  $L_i(x)$  sont données par

$$L_i(x) = \prod_{j \neq i} \frac{x - x_j}{x_i - x_j} = \frac{\omega_N(x)}{(x - x_i)\omega'_N(x_i)} \quad \text{car} \quad \omega'_N(x_i) = \prod_{j \neq i} (x_i - x_j).$$

Dans la base  $L_1, \dots, L_N$ , le polynôme  $P$  s'écrit  $P(X) = \sum_{i=1}^N y_i L_i(X)$ . On obtient donc pour  $P(x)$  la première expression de la proposition valable lorsque  $x$  est un réel distinct de  $x_1, \dots, x_N$ .

Pour la seconde, nous utilisons que  $\sum_{i=1}^N L_i(X) = 1$ . Nous avons donc

$$\sum_{i=1}^N \frac{\omega_N(x)}{(x - x_i)\omega'_N(x_i)} = 1 \quad \implies \quad \omega_N(x) = \frac{1}{\sum_{i=1}^N \frac{1}{(x - x_i)\omega'_N(x_i)}},$$

ce qui termine la preuve. ◻

Cette formule de Lagrange permet d'évaluer une valeur  $P(x)$  en un point  $x$  en  $\mathcal{O}(2N^2)$  itérations pour le premier  $x$ , puis  $\mathcal{O}(2N)$  itérations pour les suivants. En effet, le calcul de  $\omega'_N(x_i) = \prod_{j \neq i} (x_i - x_j)$  pour chaque valeur de  $i$  prend  $2N - 2$  opérations. On peut d'ailleurs stocker ces valeurs si l'on doit calculer une autre valeur de  $P(x)$  puisque cette valeur ne dépend pas de  $x$ . Pour compléter le calcul de  $P(x)$ , il ne reste plus qu'à faire 2 multiplications pour chaque valeur de  $i$  et sommer les différents termes.

Cependant, pour un traitement vectoriel (et en python, c'est le plus efficace) de cette formule, il est nécessaire de traiter séparément les points d'interpolation puisque la formule contient des divisions par 0 dans ce cas.

### ALGORITHME 2.3 – InterLagrange

```
import numpy as np
import itertools as it
def InterLagrange(x, y, xx):
    """
    Interpolation using the Lagrange formula
    """
    x = np.asarray(x, dtype = 'float')
    y = np.asarray(y, dtype = 'float')
    xx = np.asarray(xx, dtype = 'float')
    n = x.size
    if y.size != n:
        print("Error in InterLagrange: x and y don't have the same size")
    w = np.ones((n,))
    for i in range(n):
        for j in it.chain(range(i), range(i+1, n)):
            w[i] *= (x[i] - x[j])
    weight = 1./w
    if xx.size == 1:
        N, D = 0., 0.
        for i in range(n):
            if xx == x[i]:
                return y[i]
            dxi = weight[i] / (xx-x[i])
            N += y[i] * dxi
            D += dxi
    else:
        N, D = np.zeros(xx.shape), np.zeros(xx.shape)
        ind = []
        for i in range(n):
            ind.append(np.where(xx == x[i])) # indices where P(xi) = yi
            indloc = np.where(xx != x[i])
            dxi = weight[i] / (xx[indloc] - x[i]) # avoid divide by 0
            N[indloc] += y[i] * dxi
            D[indloc] += dxi
        for i in range(n): # fix yi in xi
            N[ind[i]], D[ind[i]] = y[i], 1.
    return N / D
```

La fonction `InterLagrange` prend trois arguments de type `numpy array` : `x`, `y` et `xx` et calcule le polynôme interpolateur aux points  $(x_i, y_i)_{1 \leq i \leq N}$  puis l'évalue aux points `xx` en utilisant la formule de Lagrange.

3.4 COMPORTEMENT ASYMPTOTIQUE LORSQUE  $N$  TEND VERS L'INFINI

On peut montrer que si  $f$  est une fonction définie par une série entière de rayon de convergence infini (par exemple  $f(x) = \exp(x)$  ou  $f(x) = \cos(x)$ ), le polynôme d'interpolation de Lagrange de  $f$  aux points  $x_1, x_2, \dots, x_N$  de l'intervalle  $[a, b]$  converge uniformément vers  $f$  sur  $[a, b]$  lorsque  $N$  tend vers l'infini, quel que soit la répartition des points d'interpolation.

Lorsque les points  $x_1, \dots, x_N$  sont équirépartis sur  $[a, b]$  (i.e. pour tout  $j \in \llbracket 1, N \rrbracket$ ,  $x_j = a + (j-1)h$  où  $h = (b-a)/(N-1)$ ), on pourrait s'attendre à ce que le polynôme d'interpolation de Lagrange d'une fonction  $f$  définie sur  $[a, b]$  converge uniformément vers  $f$  sur  $[a, b]$  quand  $N$  tend vers l'infini, au moins dans le cas d'une fonction assez régulière. Or dans chacun des cas suivants (et ce ne sont que des exemples)

$$f : \begin{cases} [0, 1] & \longrightarrow & \mathbb{R} \\ x & \longmapsto & \sqrt{x} \end{cases} \quad f : \begin{cases} [-1, 1] & \longrightarrow & \mathbb{R} \\ x & \longmapsto & |x| \end{cases} \quad f : \begin{cases} [-1, 1] & \longrightarrow & \mathbb{R} \\ x & \longmapsto & \frac{1}{4x^2+1} \end{cases}$$

si  $x_1, \dots, x_N$  sont équirépartis, on a  $\max |P(t) - f(t)| \rightarrow +\infty$  lorsque  $N$  tend vers l'infini. Ce comportement est appelé phénomène de Runge.

Un moyen de limiter ce phénomène est d'utiliser d'autres points d'interpolation. Si  $f$  est lipschitzienne sur  $[-1, 1]$  ou plus généralement si  $f$  est höldérienne sur  $[-1, 1]$  (i.e. il existe  $\alpha \in ]0, 1]$  et  $C > 0$  tels que  $|f(x) - f(y)| \leq C|x - y|^\alpha$  pour tout  $x, y$  dans  $[-1, 1]$ ) et si on prend pour points d'interpolation les zéros  $x_1, \dots, x_N$  du  $N$ -ième polynôme de Tchebychev, alors  $\max |P(t) - f(t)| \rightarrow 0$  lorsque  $N$  tend vers l'infini.

**DÉFINITION 2.11 – Polynômes de Tchebychev**

Les *polynômes de Tchebychev* sont définis par récurrence

$$T_0 = 1, \quad T_1 = X, \quad T_n = 2XT_{n-1} - T_{n-2}, \quad n \geq 2.$$

**PROPRIÉTÉ 2.12**

Le  $n$ ième polynôme de Tchebychev  $T_n$  vérifie les assertions suivantes :

1.  $T_n$  est de degré exactement  $n$  et son coefficient de plus haut degré vaut  $2^{n-1}$  pour  $n \geq 1$ .
2.  $T_n$  est scindé à racines simples.

$$T_n(x) = 0 \iff x \in \{x_1, \dots, x_n\}, \quad x_j = \cos\left(\frac{(2j-1)\pi}{2n}\right), \quad 1 \leq j \leq n.$$

3.  $|T_n(x)| \leq 1$  pour tout  $x \in [-1, 1]$ . De plus

$$|T_n(x)| = 1 \iff x \in \{x'_0, \dots, x'_n\}, \quad x'_k = \cos\left(\frac{k\pi}{n}\right), \quad 0 \leq k \leq n.$$

*Démonstration.* Nous démontrons le premier point par récurrence. Nous définissons  $(p_n)$  la propriété suivante : «  $T_n$  est de degré  $n$  et son coefficient de plus haut degré vaut  $2^{n-1}$  pour  $n \geq 1$  ». Les propriétés  $(p_0)$  et  $(p_1)$  sont vraies. Supposons que les propriétés  $(p_k)$  sont vraies pour  $0 \leq k \leq n$  et montrons que  $(p_{n+1})$  est vraie. Nous avons  $T_{n+1} = 2XT_n - T_{n-1}$ . Ainsi  $(p_{n+1})$  est vraie. L'essentiel de la suite de la preuve consiste à remarquer que, pour  $x \in [-1, 1]$ , la fonction polynomiale  $T_n(x)$  coïncide avec une fonction trigonométrique  $\cos(n \arccos(x))$ . Cette propriété est vraie pour  $n = 0$  et  $n = 1$  puisque  $\cos(0) = 1$  et  $\cos(\arccos(x)) = x$  si  $x \in [-1, 1]$ . Puis nous utilisons la relation de

réurrence :

$$\begin{aligned}\cos((n+1)\arccos(x)) &= \cos(n\arccos(x))\cos(\arccos(x)) - \sin(n\arccos(x))\sin(\arccos(x)), \\ \cos((n-1)\arccos(x)) &= \cos(n\arccos(x))\cos(\arccos(x)) + \sin(n\arccos(x))\sin(\arccos(x)), \\ \cos((n+1)\arccos(x)) &= 2\cos(n\arccos(x))\cos(\arccos(x)) - \cos((n-1)\arccos(x)), \\ &= 2x\cos(n\arccos(x)) - \cos((n-1)\arccos(x)).\end{aligned}$$

Ainsi, la suite de fonctions  $x \mapsto \cos(n\arccos(x))$  vérifie la même relation de récurrence que  $T_n(x)$ , les deux fonctions coïncident sur  $[-1, 1]$ . Ainsi, pour  $n \geq 1$  et pour  $x \in [-1, 1]$ ,

$$\begin{aligned}T_n(x) = 0 &\iff n\arccos(x) = \frac{\pi}{2} \pmod{\pi}, \\ &\iff \arccos(x) = \frac{\pi}{2n} \pmod{\frac{\pi}{n}}, \\ &\iff x \in \left\{ \cos\left(\frac{(2j-1)\pi}{2n}\right), j \in \llbracket 1, n \rrbracket \right\}.\end{aligned}$$

En utilisant les propriétés de la fonction  $\cos$ , nous obtenons immédiatement que  $|T_n(x)| \leq 1$  pour tout  $x \in [-1, 1]$ . De plus

$$\begin{aligned}|T_n(x)| = 1 &\iff n\arccos(x) = 0 \pmod{\pi}, \\ &\iff \arccos(x) = 0 \pmod{\frac{\pi}{n}}, \\ &\iff x \in \left\{ \cos\left(\frac{k\pi}{n}\right), k \in \llbracket 0, n \rrbracket \right\},\end{aligned}$$

ce qui termine la preuve. ◻

### PROPOSITION 2.13

Si  $Q_n$  est un polynôme de degré  $n$  et de même coefficient de plus haut degré que celui de  $T_n$ , alors

$$\max_{x \in [-1, 1]} |Q_n(x)| \geq \max_{x \in [-1, 1]} |T_n(x)| = 1.$$

*Démonstration.* Soit  $Q$  un polynôme de degré  $n \geq 1$  dont le coefficient de degré  $n$  vaut  $2^{n-1}$  (le cas  $n = 0$  est trivial). Supposons que

$$\max_{x \in [-1, 1]} |Q(x)| < \max_{x \in [-1, 1]} |T_n(x)| = 1.$$

Nous avons  $T_n(x'_k) = (-1)^k$  pour tout  $k \in \llbracket 0, n \rrbracket$ . Le polynôme  $Q_n - T_n$  admet donc au moins une racine dans l'intervalle  $]x'_{k-1}, x'_k[$  pour chaque  $k \in \llbracket 1, n \rrbracket$ . Mais c'est impossible car  $Q_n - T_n$  est un polynôme non nul de degré  $\leq n - 1$ . ◻

### COROLLAIRE 2.14

Si  $\xi_1, \dots, \xi_N$  sont  $N$  points deux à deux distincts de  $[-1, 1]$ , on a

$$\max_{x \in [-1, 1]} \left| \prod_{j=1}^N (x - \xi_j) \right| \geq \max_{x \in [-1, 1]} \left| \prod_{j=1}^N (x - x_j) \right| = \max_{x \in [-1, 1]} \frac{1}{2^{N-1}} |T_N(x)| = \frac{1}{2^{N-1}}.$$

Revenons à l'étude des polynômes interpolateurs de Lagrange. Considérons  $f$  une fonction suffisamment régulière et définissons  $P$  son polynôme interpolateur en  $N$  points de  $[-1, 1]$  notés  $\xi_1, \dots, \xi_N$ . D'après le Corollaire 2.8, l'erreur d'interpolation vérifie

$$|f(x) - P(x)| \leq \frac{\|f^{(N)}\|_\infty}{N!} |\omega_N(x)| \leq \frac{\|f^{(N)}\|_\infty}{N!} \|\omega_N\|_\infty, \quad \omega_N(x) = \prod_{j=1}^N (x - \xi_j),$$

où la norme infinie est à prendre sur l'intervalle  $[-1, 1]$ . Ainsi, le choix des points  $\xi_i$  qui fournit



une valeur de  $\|\omega_N\|_\infty$  la plus petite possible (on ne peut pas descendre sous  $2^{-N+1}$ ) est celui des zéros du polynôme de Tchebychev.

Evidemment, le cas particulier de l'intervalle  $[-1, 1]$  est facilement généralisable. En effet, si l'intervalle d'interpolation vaut  $[a, b]$ , les points d'interpolation associés sont

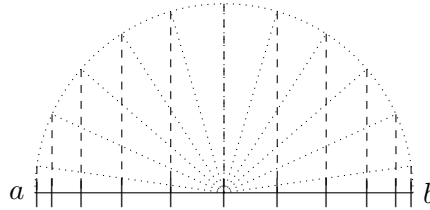
$$x_j = \frac{a+b}{2} + \frac{b-a}{2} \cos\left(\frac{(2j-1)\pi}{2N}\right), \quad 1 \leq j \leq N.$$

points équidistants



$$x_{i,N} = (i-1) \frac{b-a}{N-1}$$

points de Tchebychev



$$x_{i,N} = \frac{a+b}{2} + \frac{b-a}{2} \cos\left(\frac{(2i-1)\pi}{2N}\right) \quad 1 \leq i \leq N$$

On observe que, si la convergence semble bien fonctionner au centre de l'intervalle dans le cas des points équirépartis, de fortes oscillations de plus en plus violentes apparaissent au bord du domaine : il n'y a pas de convergence uniforme de  $P_N$  vers  $f$  dans ce cas. En revanche, avec les points de Tchebychev, la convergence semble bien uniforme. L'évolution de la norme infinie de l'erreur lorsque  $N$  augmente est donnée à la figure 2.5.

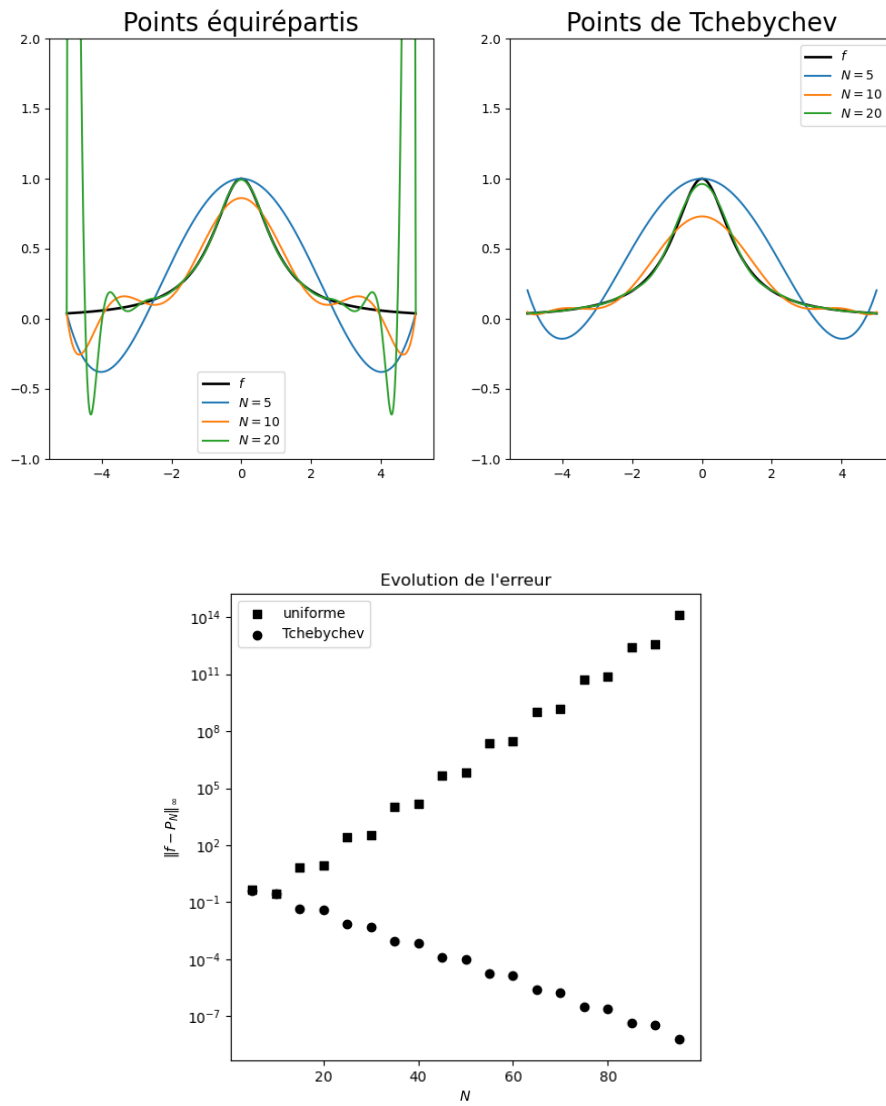


FIGURE 2.5 – Illustration du phénomène de Runge pour  $x \mapsto 1/(1+x^2)$ .

# 3 Intégration numérique

Étant donnée une fonction  $f : [a, b] \rightarrow \mathbb{R}$ , le calcul de son intégrale

$$I(f) = \int_a^b f(x) \, dx$$

est un problème qui est souvent compliqué. Dans le cas où l'expression d'une primitive  $F$  de  $f$  est connue, le problème est réglé puisque  $I(f) = F(b) - F(a)$ . Malheureusement, cela n'est pas toujours possible :

▷  $F$  peut ne pas avoir d'expression analytique, par exemple dans les cas suivants

$$x \mapsto \exp(-x^2), \quad x \mapsto \sqrt{1 + 2 \cos x}, \quad x \mapsto \frac{1}{\log x};$$

▷  $f$  peut n'être connue que par ses valeurs  $f(x)$  en certains points  $x$  (issues de données expérimentales par exemple) ;

▷  $f$  peut n'être connue que comme valeur retournée par un algorithme coûteux permettant de la calculer pour tout  $x$ .

Dans tous ces cas-là, il est nécessaire de calculer une approximation numérique  $I_{\text{app}}$  de  $I(f)$ , ce à quoi on va s'intéresser dans tout ce chapitre.

## 1 FORMULES DE QUADRATURES

Comment calculer une valeur approchée de  $I(f) = \int_a^b f(x) \, dx$  ? Un résultat théorique donne une piste : on peut recourir à des *sommes de Riemann*. Si  $\Sigma$  est une subdivision  $a = x_0 < x_1 < x_2 < \dots < x_n = b$  pointée en chaque sous-intervalle par  $x_{j+1/2} \in [x_j, x_{j+1}]$  pour tout  $i \in \llbracket 0, n-1 \rrbracket$ , on note

$$S(f, \Sigma) = \sum_{j=0}^{n-1} f(x_{j+1/2})(x_{j+1} - x_j)$$

la somme de Riemann de  $f$  associée à  $\Sigma$ , et  $\delta(\Sigma) = \max_{0 \leq j \leq n-1} |x_{j+1} - x_j|$  le pas maximal de la subdivision.

### THÉORÈME 3.1 – Sommes de Riemann

Soit  $f \in \mathcal{C}^0([a, b])$ . Étant donnée une suite de subdivisions  $\Sigma_n$  de  $[a, b]$  dont le pas maximal  $\delta(\Sigma_n)$  tend vers 0 lorsque  $n \rightarrow \infty$ , on a

$$S(f, \Sigma_n) \xrightarrow{n \rightarrow +\infty} \int_a^b f(x) \, dx.$$

Remarquons tout d'abord que le calcul de  $I_n(f) = S(f, \Sigma_n)$  n'utilise que des évaluations de  $f$  en certains points  $x_{i+1/2}$  et des opérations arithmétiques standards (sommes et produits) : on peut donc effectuer ce calcul "à la main" ou sur machine.

Néanmoins, ce théorème ne quantifie pas à quelle vitesse ces sommes de Riemann convergent vers l'intégrale de  $f$ . En réalité, si l'on prend une subdivision  $\Sigma_n$  régulièrement espacée en  $n$

sous-intervalles, ce qui correspond à  $x_j = a + jh$  où  $h = (b - a)/n$  et  $x_{j+1/2} = x_j$  par exemple, et si  $f$  est de classe  $\mathcal{C}^1$ , on ne peut s'attendre en général qu'à une convergence en  $O(1/n)$ , au sens où

$$|I(f) - I_n(f)| \leq \frac{C}{n}$$

pour une certaine constante  $C > 0$ . C'est-à-dire que pour obtenir une précision à  $10^{-6}$  de  $I(f)$  il faudrait évaluer  $f(x)$  de l'ordre de 1 million de fois, et effectuer autant d'opérations arithmétiques : c'est très coûteux et surtout, il risque d'y avoir une forte accumulation d'erreurs d'arrondis. La question est donc la suivante : peut-on trouver d'autres procédés plus efficaces pour approcher l'intégrale ?

## 1.1 DÉFINITIONS ET PREMIÈRES PROPRIÉTÉS

On va s'intéresser à des approximations d'une intégrale par des formules faisant intervenir uniquement des combinaisons linéaires de valeurs prises par  $f$  en certains points. Nous introduisons une nouvelle notation pour l'intégrale que nous souhaitons calculer afin d'éviter les confusions dans la suite de ce cours. Dans ce paragraphe, nous nous intéressons donc à évaluer

$$I = \int_{\alpha}^{\beta} \varphi(\xi) \, d\xi,$$

pour  $\varphi : [\alpha, \beta] \rightarrow \mathbb{R}$  une fonction continue.

### DÉFINITION 3.2 – Formule de quadrature

Étant donnés  $N$  points  $\xi_1, \dots, \xi_N$  dans un intervalle  $[\alpha, \beta]$  et  $N$  poids  $\omega_1, \dots, \omega_N \in \mathbb{R}$  associés à chaque point, on appelle *formule de quadrature* associée aux  $(\xi_i)_{1 \leq i \leq N}$  et  $(\omega_i)_{1 \leq i \leq N}$  l'application linéaire  $\tilde{I} : \mathcal{C}^0([\alpha, \beta]) \rightarrow \mathbb{R}$  définie par

$$\tilde{I}(\varphi) = \sum_{i=1}^N \omega_i \varphi(\xi_i).$$

### REMARQUE 3.3

En général, une formule de quadrature est donnée sous la forme

$$\int_{\alpha}^{\beta} \varphi(\xi) \, d\xi \approx \sum_{i=1}^N \omega_i \varphi(\xi_i),$$

ou est décrite par un tableau 

points	$\xi_1$	...	$\xi_N$
poids	$\omega_1$	...	$\omega_N$

.

On sait qu'au voisinage d'un point, une fonction de classe  $\mathcal{C}^{p-1}$  s'approche par un polynôme de degré  $p - 1$ . Ainsi, si une formule de quadrature approche correctement l'intégrale de certains polynômes (par exemple si elle lui est égale), elle devrait mieux approcher l'intégrale de  $\varphi$ . Cette remarque motive la définition suivante d'ordre, et on montrera plus loin dans le cours qu'elle mesure bien la qualité d'approximation de la méthode.

**DÉFINITION 3.4 – Ordre d'une méthode**

On dit qu'une formule de quadrature  $\tilde{I}$  est d'ordre  $p$  si elle est exacte pour tout polynôme de degré inférieur ou égal à  $p - 1$  (et pas plus), c'est-à-dire si

$$\begin{aligned} \forall P \in \mathbb{R}_{p-1}[X] \quad \tilde{I}(P) &= I(P), \\ \exists Q \in \mathbb{R}_p[X] : \quad \tilde{I}(Q) &\neq I(Q). \end{aligned}$$

Remarquons que, comme les applications  $\tilde{I}$  et  $I$  sont linéaires, il est équivalent de dire : la formule de quadrature  $\tilde{I}$  est d'ordre  $p$  si

$$\tilde{I}(X^k) = I(X^k), \quad 0 \leq k \leq p - 1 \quad \text{et} \quad \tilde{I}(X^p) \neq I(X^p).$$

On peut alors envisager une autre manière d'approcher l'intégrale d'une fonction  $\varphi$  en utilisant des polynômes : remplaçons  $\varphi$  par son polynôme interpolateur de Lagrange  $P_\varphi$  associé aux  $N$  points deux à deux distincts  $\xi_1, \dots, \xi_N \in [a, b]$ , c'est-à-dire

$$\int_\alpha^\beta \varphi(\xi) \, d\xi \approx \int_\alpha^\beta P_\varphi(\xi) \, d\xi = \tilde{I}(\varphi).$$

On rappelle que  $P_\varphi$  s'écrit dans la base de Lagrange composée des polynômes

$$L_i = \prod_{j \neq i} \frac{X - \xi_j}{\xi_i - \xi_j}, \quad 1 \leq i \leq N$$

sous la forme

$$P_\varphi = \sum_{i=1}^N \varphi(\xi_i) L_i.$$

De sorte que

$$\tilde{I}(\varphi) = \int_\alpha^\beta \sum_{i=1}^N \varphi(\xi_i) L_i(\xi) \, d\xi = \sum_{i=1}^N \varphi(\xi_i) \int_\alpha^\beta L_i(\xi) \, d\xi = \sum_{i=1}^N \omega_i \varphi(\xi_i),$$

où l'on a posé

$$\omega_i = \int_\alpha^\beta L_i(\xi) \, d\xi \quad 1 \leq i \leq N.$$

Cette méthode est donc un cas particulier de formule de quadrature, associée aux points  $\xi_i$  et aux poids  $\omega_i$ , qu'on qualifie de *type interpolation à  $N$  points*.

**DÉFINITION 3.5 – Formule de quadrature de type interpolation**

Etant donnés  $N$  points distincts de  $[\alpha, \beta]$  notés  $\xi_1, \dots, \xi_N$ , la *formule de quadrature de type interpolation* associée aux points  $(\xi_i)_{1 \leq i \leq N}$  est définie par

$$\tilde{I}(\varphi) = \sum_{i=1}^N \omega_i \varphi(\xi_i), \quad \text{avec} \quad \omega_i = \int_\alpha^\beta \prod_{j \neq i} \frac{\xi - \xi_j}{\xi_i - \xi_j} \, d\xi \quad 1 \leq i \leq N.$$

**PROPOSITION 3.6**

Une formule de quadrature à  $N$  points est d'ordre au moins  $N$  si, et seulement si, elle est de type interpolation à  $N$  points.

*Démonstration.* Supposons que  $\tilde{I}$  est une formule de quadrature de type interpolation aux points  $\xi_1, \dots, \xi_N$

$$\tilde{I}(\varphi) = \int_{\alpha}^{\beta} P_{\varphi}(\xi) \, d\xi,$$

où  $P_{\varphi}$  est le polynôme d'interpolation de Lagrange associé à  $\varphi$  aux points  $\xi_1, \dots, \xi_N$ . Si  $\varphi$  est un polynôme de degré inférieur ou égal à  $N - 1$ ,  $\varphi = P_{\varphi}$  et donc  $\tilde{I}(\varphi) = I(\varphi)$ , ce qui conclut.

Réciproquement soit  $\tilde{I}$  est une formule de quadrature à  $N$  points (distincts)

$$\tilde{I}(\varphi) = \sum_{i=1}^N \omega_i \varphi(\xi_i)$$

qui est exacte sur  $\mathbb{R}_{N-1}[X]$ . Considérons la base de Lagrange formée des  $L_i \in \mathbb{R}_{N-1}[X]$ ,  $1 \leq i \leq N$ , associée aux  $(\xi_i)_{1 \leq i \leq N}$ . On sait que  $\omega_j = \tilde{I}(L_j) = \int_{\alpha}^{\beta} L_j(\xi) \, d\xi$  de sorte que

$$\tilde{I}(\varphi) = \sum_{i=1}^N \omega_i \varphi(\xi_i) = \sum_{i=1}^N \int_{\alpha}^{\beta} L_i(\xi) \varphi(\xi_i) \, d\xi = \int_{\alpha}^{\beta} \sum_{i=1}^N \varphi(\xi_i) L_i(\xi) \, d\xi = \int_{\alpha}^{\beta} P_{\varphi}(\xi) \, d\xi,$$

et la formule est de type interpolation à  $N$  points. ◻

### PROPOSITION 3.7 – Erreur des méthodes de type interpolation

Si  $\tilde{I}$  est de type interpolation à  $N$  points sur  $[\alpha, \beta]$  et  $\varphi$  est de classe  $\mathcal{C}^N$  alors on a l'estimation

$$\left| I(\varphi) - \tilde{I}(\varphi) \right| \leq \frac{(\beta - \alpha)^{N+1}}{(N + 1)!} \|\varphi^{(N)}\|_{\infty}.$$

*Démonstration.* On utilise l'estimation vue au Corollaire 2.8

$$\left| \varphi(\xi) - P(\xi) \right| \leq \frac{\|\varphi^{(N)}\|_{\infty}}{N!} |\omega_N(\xi)|$$

où  $\omega_n(\xi) = (\xi - \xi_1) \dots (\xi - \xi_N)$ . En utilisant l'inégalité  $|\omega_N(\xi)| \leq (\xi - \alpha)^N$ , pour  $\alpha \leq \xi \leq \beta$ , on obtient

$$\left| I(\varphi) - \tilde{I}(\varphi) \right| = \left| \int_{\alpha}^{\beta} \varphi(\xi) - P(\xi) \, d\xi \right| \leq \int_{\alpha}^{\beta} \frac{\|\varphi^{(N)}\|_{\infty}}{N!} |\omega_N(\xi)| \, d\xi \leq \frac{(\beta - \alpha)^{N+1}}{(N + 1)!} \|\varphi^{(N)}\|_{\infty},$$

ce qui termine la preuve. ◻

Ainsi, la proposition précédente pourrait nous faire penser que, pour mieux approcher l'intégrale d'une fonction, il est suffisant d'augmenter le nombre de points  $N$  dans la formule de quadrature. Il n'en est rien comme nous l'avons vu au chapitre 2 consacré à l'interpolation : l'estimation dépendant de la dérivée  $N$ ième ne garantit pas la convergence. L'étude du phénomène de Runge a même été plus loin dans l'analyse : la différence entre la fonction et le polynôme interpolateur peut tendre vers l'infini pour certaines fonctions. Nous devons donc changer de stratégie. La figure 3.1 illustre la divergence du calcul de l'intégrale approchée en utilisant directement le polynôme interpolateur de Lagrange (avec des points équirépartis ou les points de Tchebychev). La figure a été construite en calculant l'intégrale du polynôme interpolateur pour la fonction  $x \mapsto 1/(1 + x^2)$  sur  $[0, 5]$ .

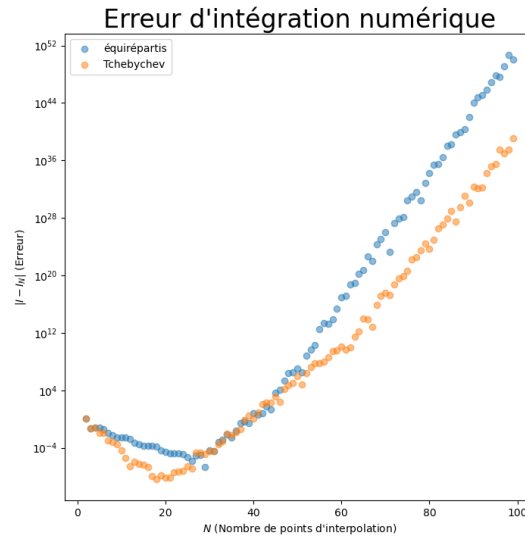


FIGURE 3.1 – Illustration de la non convergence de la méthode d'intégration par interpolation.

## 1.2 FORMULES DE QUADRATURES ÉLÉMENTAIRES ET COMPOSÉES

Soit  $f : [a, b] \rightarrow \mathbb{R}$  une fonction continue. Essayons d'approcher

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

Reprenons notre exemple précédent de somme de Riemann  $I_n(f) = S(f, \Sigma_n)$  où  $\Sigma_n$  est la subdivision régulière donnée par  $h = (b - a)/n$ ,  $x_j = a + jh$ ,  $0 \leq j \leq n$  et  $\xi_j = x_j$  pour  $0 \leq j \leq n - 1$ . On a une somme de  $n$  termes de la forme  $(x_{j+1} - x_j)f(x_j)$ , or par la relation de Chasles,  $I(f)$  s'écrit aussi comme une somme de  $n$  termes

$$I(f) = \sum_{j=0}^{n-1} \int_{x_j}^{x_{j+1}} f(x) \, dx,$$

de sorte qu'approcher  $I(f)$  par  $I_n(f)$  consiste à remplacer  $\int_{x_j}^{x_{j+1}} f(x) \, dx$  par  $(x_{j+1} - x_j)f(x_j)$ . Or lorsque  $f$  est positive,  $\int_{x_j}^{x_{j+1}} f(x) \, dx$  représente l'aire sous le graphe de  $f$  entre les abscisses  $x_j$  et  $x_{j+1}$ , et  $(x_{j+1} - x_j)f(x_j)$  représente l'aire du rectangle de même base et de hauteur  $f(x_j)$ , c'est-à-dire la valeur de la fonction à gauche de l'intervalle. On a donc découpé l'intervalle  $[a, b]$  en petits intervalles sur lesquels on a approché l'aire sous le graphe de  $f$  par une *formule de quadrature élémentaire*, appelée ici méthode des rectangles à gauche

$$\int_{x_j}^{x_{j+1}} f(x) \, dx \approx (x_{j+1} - x_j)f(x_j),$$

puis on a sommé pour obtenir une *formule de quadrature composée*

$$\int_a^b f(x) \, dx \approx \sum_{j=0}^{N-1} (x_{j+1} - x_j)f(x_j).$$

Nous allons à présent définir des méthodes de quadrature élémentaires (qui seront de type interpolation vues à la section précédente) plus précises que la méthode des rectangles à gauche afin de rester dans l'esprit des sommes de Riemann mais en améliorant la vitesse de convergence.

Remarquons tout d'abord qu'il est suffisant de définir les formules de quadratures élémentaires sur l'intervalle  $[-1, 1]$ . En effet, le changement de variables affine

$$\phi_j : \begin{cases} [-1, 1] & \longrightarrow [x_j, x_{j+1}] \\ t & \longmapsto \frac{1-t}{2}x_j + \frac{1+t}{2}x_{j+1} \end{cases} \quad (3.1)$$

permet de ramener le calcul de l'intégrale sur  $[x_j, x_{j+1}]$  à un calcul sur  $[-1, 1]$  selon la formule

$$\int_{x_j}^{x_{j+1}} f(x) \, dx = \frac{x_{j+1} - x_j}{2} \int_{-1}^1 f(\phi_j(t)) \, dt,$$

de sorte qu'il suffira d'appliquer les formules de quadrature élémentaires sur l'intervalle  $[-1, 1]$  aux fonctions  $\varphi_j = f \circ \phi_j$ .

### DÉFINITION 3.8 – Formule de quadrature élémentaire – composite

On appelle *formule de quadrature élémentaire* une formule de quadrature  $I_e$  sur  $[-1, 1]$

$$I_e(\varphi) = \sum_{i=1}^N \omega_i \varphi(\xi_i) \text{ pour } \varphi \in \mathcal{C}^0([-1, 1]).$$

Etant donnée une subdivision de l'intervalle  $[a, b]$  en  $n$  sous-intervalles de la forme

$$a = x_0 < x_1 < \dots < x_n = b,$$

la formule de quadrature élémentaire  $I_e$  induit une *formule de quadrature composite*  $I_c$  sur  $[a, b]$  qui s'écrit

$$I_c(f) = \sum_{j=0}^{n-1} \frac{x_{j+1} - x_j}{2} I_e(f \circ \phi_j) = \sum_{j=0}^{n-1} \frac{x_{j+1} - x_j}{2} \sum_{i=1}^N \omega_i f(x_{i,j}), \quad (3.2)$$

avec

$$x_{i,j} = \phi_j(\xi_i) = \frac{1 - \xi_i}{2} x_j + \frac{1 + \xi_i}{2} x_{j+1}, \quad 1 \leq i \leq N, \quad 0 \leq j \leq n - 1.$$

## 2 MÉTHODES DE QUADRATURE CLASSIQUES

Dans cette section, quelques-unes des méthodes de quadrature parmi les plus classiques seront présentées, puis analysées afin de déterminer une estimation de l'erreur commise. Commençons par appliquer la méthode générale des méthodes composites dans les cas les plus simples.

Précisons les notations. Nous noterons

$$\begin{aligned} I_0(\varphi) &= \int_{-1}^1 \varphi(\xi) \, d\xi, & I_e(\varphi) &= \sum_{i=1}^N \omega_i \varphi(\xi_i), \\ I(f) &= \int_a^b f(x) \, dx = \sum_{j=0}^{n-1} \frac{x_{j+1} - x_j}{2} I_0(f \circ \phi_j), & I_c(f) &= \sum_{j=0}^{n-1} \frac{x_{j+1} - x_j}{2} I_e(f \circ \phi_j), \end{aligned} \quad (3.3)$$



Nom	points d'interpolation	formule composite
rectangle à droite	1	$\frac{b-a}{n} \sum_{j=0}^{n-1} f(x_{j+1})$
rectangle à gauche	-1	$\frac{b-a}{n} \sum_{j=0}^{n-1} f(x_j)$
point milieu	0	$\frac{b-a}{n} \sum_{j=0}^{n-1} f\left(\frac{x_{j+1} + x_j}{2}\right)$
trapèze	-1, 1	$\frac{b-a}{2n} \sum_{j=0}^{n-1} f(x_j + 1) + f(x_j)$
Simpson	-1, 0, 1	$\frac{b-a}{6n} \sum_{j=0}^{n-1} f(x_j + 1) + 4f\left(\frac{x_{j+1} + x_j}{2}\right) + f(x_j)$

TABLE 3.1 – Les méthodes de quadratures classiques : méthodes composées à partir des méthodes de type interpolation.

où  $\phi_j : [-1, 1] \rightarrow [x_j, x_{j+1}]$   $t \mapsto \frac{1-t}{2}x_j + \frac{1+t}{2}x_{j+1}$  est définie en (3.1).

## 2.1 UNE LISTE DES MÉTHODES CLASSIQUES

Nous considérons ici différentes méthodes de quadrature élémentaires de type interpolation et nous précisons la formule de quadrature composite obtenue. Nous choisissons pour simplifier les formules une subdivision régulière de l'intervalle  $[a, b]$ . C'est-à-dire que nous avons  $x_i = a + ih$  avec  $h = (b - a)/n$ . Le tableau 3.1 récapitule les formules classiques de quadrature.

Détaillons à présent comment sont obtenues ces formules. Pour la *méthode des rectangles à droite*, nous avons un seul point d'interpolation : le point  $\xi_1 = 1$ . Ainsi, la fonction  $\varphi$  est remplacée par son polynôme interpolateur de Lagrange au point  $\xi_1 = 1$ . Nous pouvons écrire

$$P_\varphi(\xi) = \varphi(1) \implies I_e(\varphi) = \int_{-1}^1 \varphi(1) \, d\xi = 2\varphi(1).$$

Nous en déduisons à l'aide de la formule (3.2)

$$I_c(f) = \frac{b-a}{n} \sum_{j=0}^{n-1} f(x_{j+1}). \quad (\text{Rectangles à droite})$$

La *formule des rectangles à gauche* s'obtient exactement de la même manière en choisissant comme polynôme interpolateur le polynôme interpolateur de Lagrange au point  $\xi_1 = -1$  :

$$P_\varphi(\xi) = \varphi(-1) \implies I_e(\varphi) = \int_{-1}^1 \varphi(-1) \, d\xi = 2\varphi(-1).$$

Nous en déduisons à l'aide de la formule (3.2)

$$I_c(f) = \frac{b-a}{n} \sum_{j=0}^{n-1} f(x_j). \quad (\text{Rectangles à gauche})$$

La *formule du point milieu* est obtenue quant à elle en choisissant le polynôme interpolateur de Lagrange au point  $\xi_1 = 0$  :

$$P_\varphi(\xi) = \varphi(0) \implies I_e(\varphi) = \int_{-1}^1 \varphi(0) \, d\xi = 2\varphi(0).$$

Dans ce cas, nous avons  $x_{1,j} = (x_j + x_{j+1})/2$  et

$$I_c(f) = \frac{b-a}{n} \sum_{j=0}^{n-1} f\left(\frac{x_{j+1} + x_j}{2}\right). \quad (\text{Points milieux})$$

La *formule des trapèzes* est obtenue en choisissant le polynôme interpolateur de Lagrange aux deux points  $\xi_1 = -1$  et  $\xi_2 = 1$ . Dans ce cas, le polynôme est de degré au plus 1, ainsi

$$P_\varphi(\xi) = \frac{1-\xi}{2}\varphi(-1) + \frac{1+\xi}{2}\varphi(1) \implies I_e(\varphi) = \int_{-1}^1 P_\varphi(\xi) \, d\xi = \varphi(-1) + \varphi(1).$$

Dans ce cas, nous avons  $x_{1,j} = x_j$  et  $x_{2,j} = x_{j+1}$  et

$$I_c(f) = \frac{b-a}{2n} \sum_{j=0}^{n-1} [f(x_j) + f(x_{j+1})]. \quad (\text{Trapèzes})$$

Finalement, la *méthode de Simpson* est obtenue en choisissant le polynôme interpolateur de Lagrange aux trois points  $\xi_1 = -1$ ,  $\xi_2 = 0$  et  $\xi_3 = 1$ . Ce polynôme de degré au plus 2 s'écrit

$$P_\varphi(\xi) = \frac{\xi(\xi-1)}{2}\varphi(-1) + (1+\xi)(1-\xi)\varphi(0) + \frac{\xi(1+\xi)}{2}\varphi(1).$$

Nous avons donc (en utilisant que les contributions des parties impaires sont nulles)

$$\omega_1 = \int_{-1}^1 \frac{\xi(\xi-1)}{2} \, d\xi = \left[ \frac{\xi^3}{6} \right]_{-1}^1 = \frac{1}{3},$$

$$\omega_2 = \int_{-1}^1 (1+\xi)(1-\xi) \, d\xi = \left[ 1 - \frac{\xi^3}{3} \right]_{-1}^1 = 2 - \frac{2}{3} = \frac{4}{3},$$

$$\omega_3 = \int_{-1}^1 \frac{\xi(\xi+1)}{2} \, d\xi = \left[ \frac{\xi^3}{6} \right]_{-1}^1 = \frac{1}{3}.$$

Nous en déduisons alors la formule composée

$$I_c(f) = \frac{b-a}{6n} \sum_{j=0}^{n-1} \left[ f(x_{j+1}) + 4f\left(\frac{x_{j+1} + x_j}{2}\right) + f(x_j) \right]. \quad (\text{Simpson})$$

La figure 3.2 illustre la convergence de ces 5 méthodes d'intégration numérique. Les courbes ont été obtenues en calculant l'erreur du calcul approché avec  $N$  points ( $N$  variant entre 1 et 100) pour calculer

$$I = \int_0^5 \frac{1}{1+x^2} \, dx.$$

En échelle logarithmique, l'erreur est asymptotiquement proche d'une droite de pente entière (1 pour les méthodes des rectangles, 2 pour les méthodes des points milieux et des trapèzes et 4 pour la méthode de Simpson). Les sections suivantes vont permettre de comprendre pourquoi ces pentes sont observées.

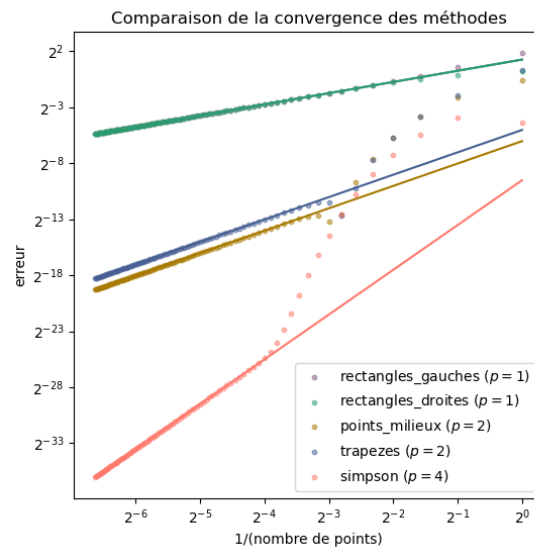


FIGURE 3.2 – Illustration de la convergence des méthodes d'intégration numérique.

## 2.2 ANALYSE DES MÉTHODES DES RECTANGLES

Considérons la formule des rectangles à gauche

$$I_e(\varphi) = 2\varphi(-1)$$

pour  $\varphi \in \mathcal{C}^0([-1, 1])$ . On cherche à calculer l'ordre de la méthode et quantifier l'erreur commise par rapport à la valeur exacte de l'intégrale. Une illustration des méthodes des rectangles à gauche et à droite se trouve à la figure 3.3.

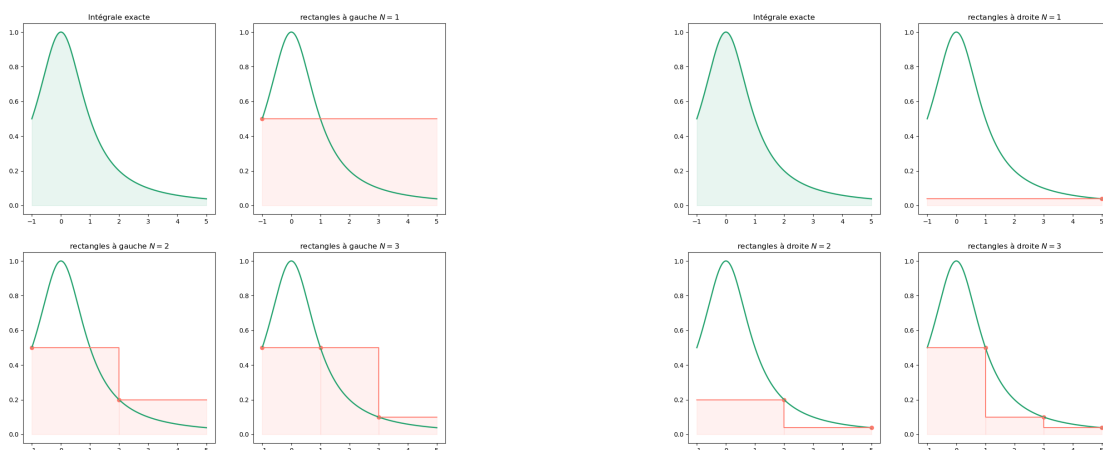


FIGURE 3.3 – Illustration de la méthode du rectangle.

### PROPOSITION 3.9

La méthode des rectangles à gauche est d'ordre 1.

*Démonstration.* On sait déjà que c'est une méthode d'interpolation à 1 point donc elle est d'ordre au moins 1. Pour montrer qu'elle n'est pas d'ordre plus grand il suffit d'exhiber un polynôme  $P$  de degré 1 tel que  $I_e(P) \neq I_0(P)$ . Le choix  $P = X$  convient :  $I_e(P) = -2$  tandis que  $I_0(P) = \int_{-1}^1 x \, dx = 0$ .  $\bullet$

**PROPOSITION 3.10 – Erreur de la méthode des rectangles à gauche**

Si  $\varphi$  est de classe  $\mathcal{C}^1$  sur  $[-1, 1]$ , il existe  $c \in ]-1, 1[$  tel que l'erreur de la méthode de quadrature élémentaire s'écrit

$$E_e(\varphi) = I_0(\varphi) - I_e(\varphi) = 2\varphi'(c).$$

Si  $f$  est de classe  $\mathcal{C}^1$  sur  $[a, b]$  l'erreur de quadrature de la méthode composite associée à une subdivision régulière de pas  $h$  est majorée par

$$|E_c(f)| = |I(f) - I_c(f)| \leq h \|f'\|_\infty \frac{b-a}{2}.$$

*Démonstration.* Soit  $\phi$  une primitive de  $\varphi$  sur l'intervalle  $[-1, 1]$ . Effectuons un développement de Taylor avec reste de Lagrange de  $\phi$  au point  $-1$  :

$$\exists c \in ]-1, 1[ : \quad \phi(1) = \phi(-1) + 2\phi'(-1) + 2\phi''(c).$$

or  $I_0(\varphi) = \phi(1) - \phi(-1)$  et  $\phi' = \varphi$ ,  $\phi'' = \varphi'$  donc

$$I_0(\varphi) - I_e(\varphi) = \int_{-1}^1 \varphi(t) \, dt - 2\varphi(-1) = 2\varphi'(c).$$

Pour  $f \in \mathcal{C}^1([a, b], \mathbb{R})$ , nous avons grâce aux formules (3.3)

$$\begin{aligned} I(f) - I_c(f) &= \sum_{j=0}^{n-1} \frac{x_{j+1} - x_j}{2} (I_0(f \circ \phi_j) - I_e(f \circ \phi_j)) \\ &= \sum_{j=0}^{n-1} (x_{j+1} - x_j) (f \circ \phi_j)'(c_j) = \sum_{j=0}^{n-1} (x_{j+1} - x_j) f'(\phi_j(c_j)) \phi_j'(c_j), \end{aligned}$$

où  $c_j \in ]-1, 1[$ ,  $0 \leq j \leq n-1$ . Comme  $\phi_j'(c_j) = (x_{j+1} - x_j)/2$ , nous obtenons

$$|I(f) - I_c(f)| \leq \sum_{j=0}^{n-1} \frac{(x_{j+1} - x_j)^2}{2} \|f'\|_\infty = h \|f'\|_\infty \frac{b-a}{2},$$

ce qui termine la preuve.  $\bullet$

La formule des rectangles à droite se traite de la même manière.

Remarquons à ce niveau que nous pouvons faire le lien entre l'ordre de la méthode de quadrature élémentaire (la propriété d'être une formule exacte pour les polynôme d'un certain degré) et l'ordre de convergence de la méthode de quadrature. Nous pouvons en effet démontrer une estimation de l'erreur en  $\mathcal{O}(h)$  sans utiliser le fait que la méthode est de type interpolation. Il est suffisant de remarquer que la méthode des rectangles (à gauche ou à droite, cela ne change rien) est une formule de quadrature d'ordre 1, c'est-à-dire que  $I_e(P) = I_0(P)$  dès que  $P \in \mathbb{R}_0[X]$ . Or nous pouvons écrire

$$\forall \xi \in [-1, 1] \quad \exists \eta \in [-1, 1] : \quad \varphi(\xi) = \varphi(0) + \varphi'(\eta)\xi.$$

En notant  $P$  le polynôme obtenu en tronquant le développement de Taylor à l'ordre le plus bas (c'est-à-dire  $P = \varphi(0)$ ), nous avons

$$\forall \xi \in [-1, 1] \quad |\varphi(\xi) - P(\xi)| \leq \|\varphi'\|_\infty |\xi|.$$

Nous en déduisons par linéarité de  $I_0$  et de  $I_e$  que

$$\begin{aligned} |I_0(\varphi) - I_e(\varphi)| &= |I_0(\varphi) - I_0(P) + I_e(P) - I_e(\varphi)|, \\ |I_0(\varphi) - I_0(P)| &\leq \int_{-1}^1 |\varphi(\xi) - P(\xi)| \, d\xi \leq \|\varphi'\|_\infty \int_{-1}^1 |\xi| \, d\xi = \|\varphi'\|_\infty, \\ |I_e(P) - I_e(\varphi)| &= 2|P(-1) - \varphi(-1)| \leq 2\|\varphi'\|_\infty. \end{aligned}$$

En reprenant la preuve de la proposition précédente, c'est cette inégalité qui permet ensuite d'obtenir la majoration  $|I(f) - I_c(f)| \leq \mathcal{O}(h)$ . Ainsi, si la formule de quadrature est d'ordre 1, l'erreur de la méthode composée associée est en  $\mathcal{O}(h)$ .

## 2.3 ANALYSE DE LA MÉTHODE DES TRAPÈZES

Considérons la formule des trapèzes

$$I_e(\varphi) = \varphi(-1) + \varphi(1)$$

pour  $\varphi \in \mathcal{C}^0([-1, 1])$ . Une illustration de la méthode des trapèzes se trouve à la figure 3.4. On cherche à calculer l'ordre de la méthode et à quantifier l'erreur commise par rapport à la valeur exacte de l'intégrale.

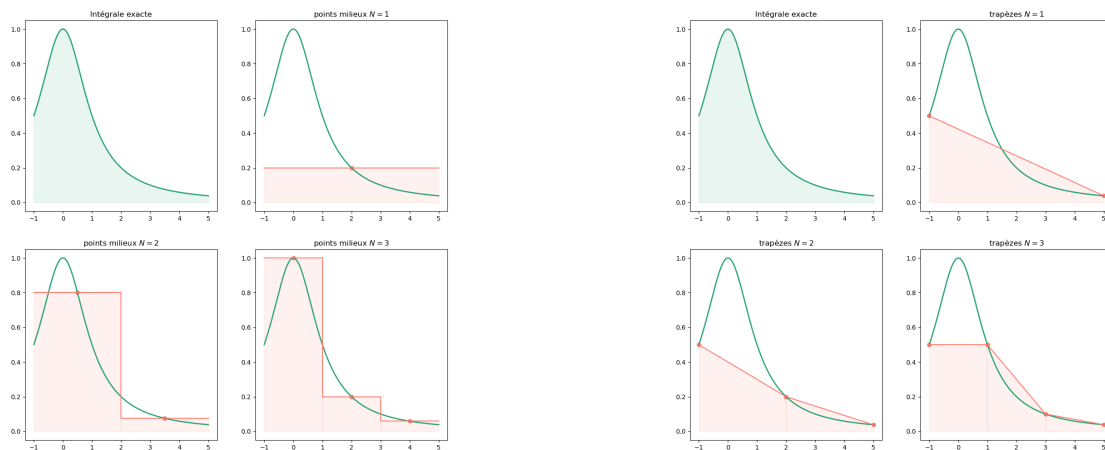


FIGURE 3.4 – Illustration de la méthode du point milieu à gauche et du trapèze à droite.

### PROPOSITION 3.11

La méthode des trapèzes est d'ordre 2.

*Démonstration.* On sait déjà que c'est une méthode d'interpolation à 2 points donc elle est d'ordre au moins 2. Pour montrer qu'elle n'est pas d'ordre plus grand, nous choisissons le polynôme  $P = X^2$ . Nous avons  $I_e(P) = 2$  et  $I_0(P) = \int_{-1}^1 x^2 \, dx = 2/3$ . ◻

**PROPOSITION 3.12 – Erreur de la méthode des trapèzes**

Si  $\varphi$  est de classe  $\mathcal{C}^2$  sur  $[-1, 1]$ , l'erreur de la méthode de quadrature élémentaire est majorée par

$$|E_e(\varphi)| = |I_0(\varphi) - I_e(\varphi)| \leq \frac{2}{3} \|\varphi''\|_\infty.$$

Si  $f$  est de classe  $\mathcal{C}^2$  sur  $[a, b]$  l'erreur de quadrature de la méthode composite associée à une subdivision régulière de pas  $h$  est majorée par

$$|E_c(f)| = |I(f) - I_c(f)| \leq h^2 \|f''\|_\infty \frac{b-a}{12}.$$

*Démonstration.* Comme la méthode des trapèzes est une méthode de type interpolation à deux points, nous appliquons la proposition 3.7 : pour  $\varphi \in \mathcal{C}^2([-1, 1])$  nous avons

$$|E_e(\varphi)| \leq \frac{2^3}{3!} \|\varphi''\|_\infty = \frac{4}{3} \|\varphi''\|_\infty.$$

Pour améliorer la constante, il est possible de reprendre la preuve de la proposition (3.7) pour écrire

$$|E_e(\varphi)| \leq \frac{1}{2!} \|\varphi''\|_\infty \int_{-1}^1 (1-x)(1+x) dx \leq \frac{1}{2} \|\varphi''\|_\infty \left[ x - \frac{x^3}{3} \right]_{-1}^1 = \frac{2}{3} \|\varphi''\|_\infty.$$

Pour  $f \in \mathcal{C}^2([a, b], \mathbb{R})$ , nous avons grâce aux formules (3.3)

$$\begin{aligned} |I(f) - I_c(f)| &\leq \sum_{j=0}^{n-1} \frac{x_{j+1} - x_j}{2} |I_0(f \circ \phi_j) - I_e(f \circ \phi_j)| \\ &\leq \frac{1}{3} \sum_{j=0}^{n-1} (x_{j+1} - x_j) \|(f \circ \phi_j)''\|_\infty. \end{aligned}$$

Comme  $\phi_j'(x) = (x_{j+1} - x_j)/2$ , nous avons

$$(f \circ \phi_j)'(\xi) = \frac{h}{2} f' \circ \phi_j(\xi), \quad (f \circ \phi_j)''(\xi) = \frac{h^2}{4} f'' \circ \phi_j(\xi).$$

Nous obtenons donc

$$|I(f) - I_c(f)| \leq \frac{1}{3} \sum_{j=0}^{n-1} \frac{(x_{j+1} - x_j)^3}{4} \|f''\|_\infty = h^2 \|f''\|_\infty \frac{b-a}{12},$$

ce qui termine la preuve. ◻

## 2.4 ANALYSE DE LA MÉTHODE DU POINT MILIEU

Considérons la formule du point milieu

$$I_e(\varphi) = 2\varphi(0)$$

pour  $\varphi \in \mathcal{C}^0([-1, 1])$ . Une illustration de la méthode des points milieux se trouve à la figure 3.4.

**PROPOSITION 3.13**

La méthode du point milieu est d'ordre 2.

*Démonstration.* Comme c'est une méthode d'interpolation à 1 point, la méthode est au moins d'ordre 1. Testons la méthode sur la base canonique des polynômes :

$$\begin{aligned} P = 1, & & I_0(P) &= \int_{-1}^1 1 \, dx = 2, & & I_e(P) = 2P(0) = 2, \\ P = X, & & I_0(P) &= \int_{-1}^1 x \, dx = 0, & & I_e(P) = 2P(0) = 0, \\ P = X^2, & & I_0(P) &= \int_{-1}^1 x^2 \, dx = \frac{2}{3}, & & I_e(P) = 2P(0) = 0. \end{aligned}$$

La méthode est donc d'ordre exactement 2. ◻

### PROPOSITION 3.14 – Erreur de la méthode du point milieu

Si  $\varphi$  est de classe  $\mathcal{C}^2$  sur  $[-1, 1]$ , il existe  $c \in ]-1, 1[$  tel que l'erreur de la méthode de quadrature élémentaire s'écrit

$$E_e(\varphi) = I_0(\varphi) - I_e(\varphi) = \frac{1}{3}\varphi''(c).$$

Si  $f$  est de classe  $\mathcal{C}^2$  sur  $[a, b]$  l'erreur de quadrature de la méthode composite associée à une subdivision régulière de pas  $h$  est majorée par

$$|E_c(f)| = |I(f) - I_c(f)| \leq h^2 \|f''\|_\infty \frac{b-a}{24}.$$

*Démonstration.* On pose  $\phi(x) = \int_{-1}^x \varphi(t) \, dt$  et on fait un développement de Taylor-Lagrange à l'ordre 3 en 0

$$\begin{aligned} \phi(-1) &= \phi(0) - \phi'(0) + \frac{1}{2}\phi''(0) - \frac{1}{6}\phi^{(3)}(c_1), \\ \phi(1) &= \phi(0) + \phi'(0) + \frac{1}{2}\phi''(0) + \frac{1}{6}\phi^{(3)}(c_2), \end{aligned}$$

pour certains  $c_1, c_2 \in ]-1, 1[$ . Or  $\int_{-1}^1 \varphi(t) \, dt = \phi(1) - \phi(-1)$  et  $\phi' = \varphi$  donc en faisant la différence :

$$\int_{-1}^1 \varphi(t) \, dt = 2\varphi(0) + \frac{1}{6}(\varphi''(c_1) + \varphi''(c_2)).$$

La fonction  $\varphi''$  étant continue, d'après le théorème des valeurs intermédiaires, il existe  $c \in ]a, b[$  tel que  $(\varphi''(c_1) + \varphi''(c_2))/2 = \varphi''(c)$ , d'où

$$\int_{-1}^1 \varphi(t) \, dt - 2\varphi(0) = \frac{1}{3}\varphi''(c).$$

En utilisant le changement de variable  $\phi_j$  sur l'intervalle  $[x_j, x_{j+1}]$ , on en déduit que si  $f$  est de classe  $\mathcal{C}^2$  sur  $[a, b]$ ,

$$\begin{aligned} I(f) - I_c(f) &= \sum_{j=0}^{n-1} \frac{x_{j+1} - x_j}{2} (I_0(f \circ \phi_j) - I_e(f \circ \phi_j)) \\ &= \sum_{j=0}^{n-1} \frac{x_{j+1} - x_j}{6} (f \circ \phi_j)''(c_j) = \sum_{j=0}^{n-1} \frac{(x_{j+1} - x_j)^3}{24} f^{(2)}(\phi_j(c_j)). \end{aligned}$$

Donc

$$|I(f) - I_c(f)| \leq h^2 \|f''\|_\infty \frac{b-a}{24}$$

ce qui termine la preuve. ◻

## 2.5 ANALYSE DE LA MÉTHODE DE SIMPSON

Considérons la formule de Simpson

$$I_e(\varphi) = \frac{1}{3}\varphi(-1) + \frac{4}{3}\varphi(0) + \frac{1}{3}\varphi(1)$$

pour  $\varphi \in \mathcal{C}^0([-1, 1])$ . Une illustration de la méthode de Simpson se trouve à la figure 3.5.

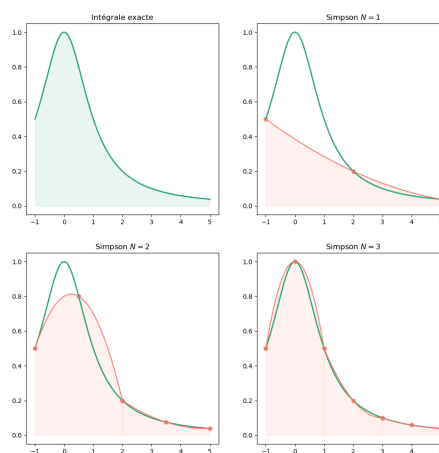


FIGURE 3.5 – Illustration de la méthode de Simpson.

**PROPOSITION 3.15**

La méthode de Simpson est d'ordre 4.

*Démonstration.* Comme c'est une méthode d'interpolation à 3 point, la méthode est au moins d'ordre 3. Testons la méthode sur la base canonique des polynômes :

$$P = 1, \quad I_0(P) = \int_{-1}^1 1 \, dx = 2, \quad I_e(P) = \frac{1}{3}P(-1) + \frac{4}{3}P(0) + \frac{1}{3}P(1) = 2,$$

$$P = X, \quad I_0(P) = \int_{-1}^1 x \, dx = 0, \quad I_e(P) = \frac{1}{3}P(-1) + \frac{4}{3}P(0) + \frac{1}{3}P(1) = 0,$$

$$P = X^2, \quad I_0(P) = \int_{-1}^1 x^2 \, dx = \frac{2}{3}, \quad I_e(P) = \frac{1}{3}P(-1) + \frac{4}{3}P(0) + \frac{1}{3}P(1) = \frac{2}{3},$$

$$P = X^3, \quad I_0(P) = \int_{-1}^1 x^3 \, dx = 0, \quad I_e(P) = \frac{1}{3}P(-1) + \frac{4}{3}P(0) + \frac{1}{3}P(1) = 0,$$

$$P = X^4, \quad I_0(P) = \int_{-1}^1 x^4 \, dx = \frac{2}{5}, \quad I_e(P) = \frac{1}{3}P(-1) + \frac{4}{3}P(0) + \frac{1}{3}P(1) = \frac{2}{3}.$$

La méthode est donc d'ordre exactement 4. Ⓜ



**PROPOSITION 3.16 – Erreur de la méthode de Simpson**

Si  $\varphi$  est de classe  $\mathcal{C}^4$  sur  $[-1, 1]$ , l'erreur de la méthode de quadrature élémentaire est majorée par

$$|E_e(\varphi)| = |I_0(\varphi) - I_e(\varphi)| \leq \frac{2}{45} \|\varphi^{(4)}\|_\infty.$$

Si  $f$  est de classe  $\mathcal{C}^4$  sur  $[a, b]$  l'erreur de quadrature de la méthode composite associée à une subdivision régulière de pas  $h$  est majorée par

$$|E_c(f)| = |I(f) - I_c(f)| \leq h^4 \|f^{(4)}\|_\infty \frac{b-a}{720}.$$

*Démonstration.* On pose  $\phi(x) = \int_{-1}^x \varphi(t) dt$  et on fait un développement de Taylor-Lagrange à l'ordre 5 en 0

$$\begin{aligned} \phi(-1) &= \phi(0) - \phi'(0) + \frac{1}{2}\phi''(0) - \frac{1}{6}\phi^{(3)}(0) + \frac{1}{24}\phi^{(4)}(0) - \frac{1}{120}\phi^{(5)}(c_1), \\ \phi(1) &= \phi(0) + \phi'(0) + \frac{1}{2}\phi''(0) + \frac{1}{6}\phi^{(3)}(0) + \frac{1}{24}\phi^{(4)}(0) + \frac{1}{120}\phi^{(5)}(c_2), \end{aligned}$$

pour certains  $c_1, c_2 \in ]-1, 1[$ . Ainsi

$$I_0(\varphi) = \phi(1) - \phi(-1) = 2\varphi(0) + \frac{1}{3}\varphi''(0) + \frac{1}{120}(\varphi^{(4)}(c_1) + \varphi^{(4)}(c_2)).$$

Nous effectuons alors un développement de Taylor-Lagrange à l'ordre 4 de la fonction  $\varphi$  en 0 également

$$\begin{aligned} \varphi(-1) &= \varphi(0) - \varphi'(0) + \frac{1}{2}\varphi''(0) - \frac{1}{6}\varphi^{(3)}(0) + \frac{1}{24}\varphi^{(4)}(c_3), \\ \varphi(1) &= \varphi(0) + \varphi'(0) + \frac{1}{2}\varphi''(0) + \frac{1}{6}\varphi^{(3)}(0) + \frac{1}{24}\varphi^{(4)}(c_4), \end{aligned}$$

pour certains  $c_3, c_4 \in ]-1, 1[$ . Ainsi

$$I_e(\varphi) = 2\varphi(0) + \frac{1}{3}\varphi''(0) + \frac{1}{72}(\varphi^{(4)}(c_3) + \varphi^{(4)}(c_4)).$$

Nous en déduisons que

$$|E_e(\varphi)| = |I_0(\varphi) - I_e(\varphi)| \leq \left(\frac{2}{120} + \frac{2}{72}\right) \|\varphi^{(4)}\|_\infty = \frac{2}{45} \|\varphi^{(4)}\|_\infty.$$

En utilisant le changement de variable  $\phi_j$  sur l'intervalle  $[x_j, x_{j+1}]$ , on en déduit que si  $f$  est de classe  $\mathcal{C}^4$  sur  $[a, b]$ ,

$$\begin{aligned} I(f) - I_c(f) &= \sum_{j=0}^{n-1} \frac{x_{j+1} - x_j}{2} (I_0(f \circ \phi_j) - I_e(f \circ \phi_j)) \\ &= \sum_{j=0}^{n-1} \frac{x_{j+1} - x_j}{45} (f \circ \phi_j)^{(4)}(c_j) = \sum_{j=0}^{n-1} \frac{(x_{j+1} - x_j)^5}{720} f^{(4)}(\phi_j(c_j)). \end{aligned}$$

Donc

$$|I(f) - I_c(f)| \leq h^4 \|f^{(4)}\|_\infty \frac{b-a}{720}$$

ce qui termine la preuve. ◻

## 2.6 CONTRÔLE DE L'ERREUR

En utilisant les théorèmes de majoration de l'erreur de quadrature, il est possible de construire un maillage de l'intervalle  $[a, b]$  afin de garantir que l'erreur soit inférieure à un certain seuil  $\epsilon$ .

Supposons que nous aillons à disposition une formule de quadrature d'ordre  $k$ , c'est-à-dire qu'elle vérifie (avec les notations utilisées dans cette section)

$$|E_e(\varphi)| \leq C \|\varphi^{(k)}\|_\infty$$

pour une certaine constante  $C$ . Si l'on considère une subdivision de l'intervalle  $[a, b]$  non nécessairement régulière  $x_0 = a < x_1 < \dots < x_{n-1} < x_n = b$ , l'erreur globale de la méthode est alors majorée par

$$|I(f) - I_c(f)| \leq C(b-a) \sum_{j=0}^{n-1} \frac{(x_{j+1} - x_j)^k}{2^k} \sup_{t \in [x_j, x_{j+1}]} |f^{(k)}(t)|.$$

Il est ainsi possible de choisir une subdivision telle que chacun des termes de la somme soit inférieur à un certain  $\epsilon/n$ , ce qui implique que l'erreur globale sera inférieure à ce  $\epsilon$ .

La figure 3.6 permet de visualiser pour la fonction  $x \mapsto e^{-x^2} - e^{-25x^2}$  que l'on intègre sur l'intervalle  $[-4, 4]$  des maillages construits pour que chaque méthode produise une erreur inférieure à  $10^{-3}$ .

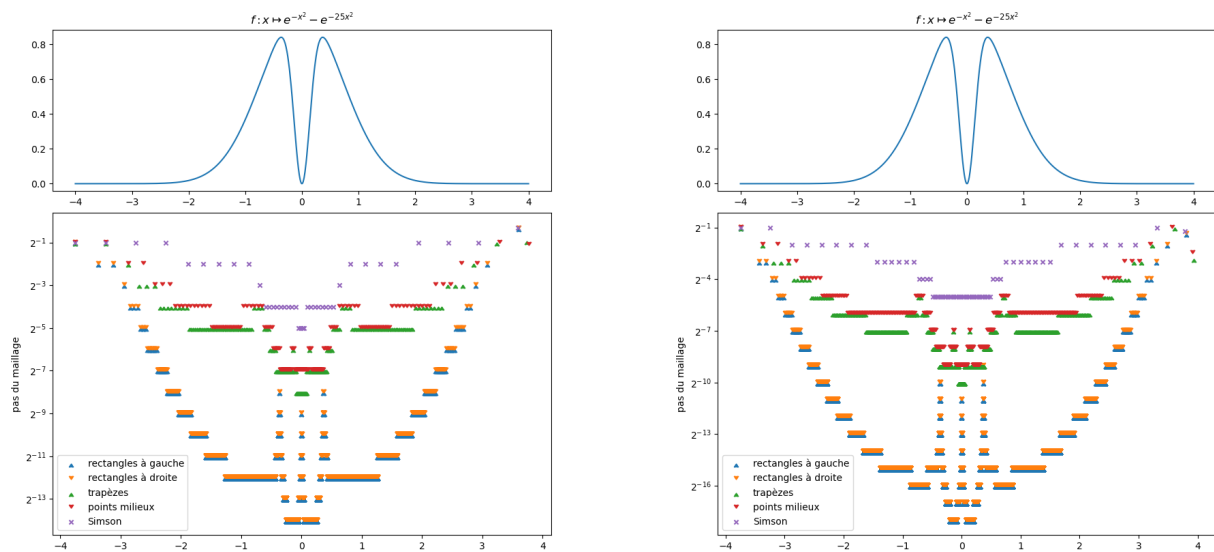


FIGURE 3.6 – Illustration de maillages adaptés pour les différentes méthodes : à gauche  $\epsilon = 10^{-3}$ , à droite  $\epsilon = 10^{-4}$ .

Dans le tableau 3.2, nous avons affiché l'erreur de quadrature qui est effectivement inférieure à  $10^{-3}$  et le nombre de points du maillage. On observe que les méthodes d'ordre 1 (les méthodes des rectangles) nécessitent beaucoup plus de points que les méthodes d'ordre plus élevé : plus de 17000 alors que la méthode de Simpson d'ordre 4 seulement 41. Lorsque le seuil est diminué (dans le tableau 3.3,  $\epsilon = 10^{-4}$ ), la conclusion est encore plus flagrante. Grossièrement, pour diviser le seuil par 10 avec une méthode d'ordre 1, il faut environ 10 fois plus de points. Pour une méthode d'ordre plus élevé, la croissance est beaucoup plus lente.

La conclusion est que (lorsque la fonction que l'on intègre est régulière), il est préférable d'utiliser des méthodes d'ordre élevé. Même si chaque calcul coûte plus cher (l'évaluation de la méthode de Simpson nécessite environ 2 fois plus de calcul que les autres méthodes), comme elle nécessite un maillage beaucoup plus grossier, elle est avantageuse.

<i>Nom de la méthode</i>	erreur	nombre de points dans le maillage
<i>rectangle à droite</i>	$3.323E-05$	17693
<i>rectangle à gauche</i>	$8.314E-06$	17693
<i>point milieu</i>	$2.198E-04$	250
<i>trapèze</i>	$2.245E-04$	189
<i>Simpson</i>	$3.233E-05$	41

TABLE 3.2 – Illustration de l’adaptation du maillage pour un seuil  $\epsilon = 10^{-3}$ 

<i>Nom de la méthode</i>	erreur	nombre de points dans le maillage
<i>rectangle à droite</i>	$1.982E-06$	203638
<i>rectangle à gauche</i>	$1.268E-06$	203638
<i>point milieu</i>	$2.759E-05$	831
<i>trapèze</i>	$2.895E-05$	569
<i>Simpson</i>	$2.407E-06$	69

TABLE 3.3 – Illustration de l’adaptation du maillage pour un seuil  $\epsilon = 10^{-4}$ 

### 3 LES ALGORITHMES DES MÉTHODES DE QUADRATURE CLASSIQUES

Dans cette section, nous proposons une version python des algorithmes qui ont été étudiés dans ce chapitre. Ils n’ont pas la prétention d’être parfait... et sont limités à des maillages uniformes, c’est-à-dire que les points sont définis par  $x_i = a + ih$  avec  $h = (b - a)/n$ .

Toutes les fonctions proposées prennent quatre arguments : la fonction **f** à intégrer, deux réels **a** et **b** qui définissent le domaine d’intégration et un entier **N** qui détermine le nombre de découpage pour la méthode composée.

#### ALGORITHME 3.1 – Rectangles à gauche

```
def rectangles_gauche(f, a, b, N):
    x, h = np.linspace(a, b, N+1, retstep=True)
    xl = x[:-1]
    return h*np.sum(f(xl))
```

#### ALGORITHME 3.2 – Rectangles à droite

```
def rectangles_droite(f, a, b, N):
    x, h = np.linspace(a, b, N+1, retstep=True)
    xr = x[1:]
    return h*np.sum(f(xr))
```

#### ALGORITHME 3.3 – Trapèzes

```
def trapezes(f, a, b, N):
    x, h = np.linspace(a, b, N+1, retstep=True)
    xl, xr = x[:-1], x[1:]
    return .5*h*(np.sum(f(xl)) + np.sum(f(xr)))
```

ALGORITHME 3.4 – *Points milieux*

```
def points_milieux(f, a, b, N):  
    x, h = np.linspace(a, b, N+1, retstep=True)  
    xm = .5*(x[:-1]+x[1:])  
    return h*np.sum(f(xm))
```

ALGORITHME 3.5 – *Simpson*

```
def simpson(f, a, b, N):  
    x, h = np.linspace(a, b, N+1, retstep=True)  
    xl, xr = x[:-1], x[1:]  
    xm = .5*(xl+xr)  
    return h/6*(  
        np.sum(f(xl)) + 4*np.sum(f(xm)) + np.sum(f(xr))  
    )
```

# 4 Résolution d'équations ordinaires

Ce chapitre est consacré à la recherche de solution approchée d'équations non linéaires scalaires. Soit  $I$  un intervalle de  $\mathbb{R}$  et soit  $f : I \rightarrow \mathbb{R}$  continue. On considère le problème suivant

$$\text{Chercher } x^* \in I \text{ tel que } f(x^*) = 0. \quad (4.1)$$

## DÉFINITION 4.1

Toute solution  $x^* \in I$  de (4.1) est dite *racine* ou *zéro* de  $f$  dans  $I$ .

Si  $f$  est de classe  $C^r$  sur  $I$  avec  $r \geq 1$  et  $x^*$  une racine de  $f$  dans  $I$ , alors

- ▷  $x^*$  est dite *racine simple* si  $f'(x^*) \neq 0$ ,
- ▷  $x^*$  est dite *racine de multiplicité*  $p < r$ , si  $f^{(k)}(x^*) = 0$  pour  $0 \leq k < p$  et  $f^{(p)}(x^*) \neq 0$ .

## 1 EXEMPLES D'APPLICATION

Nous présentons deux exemples classiques pour lesquels il est nécessaire de résoudre de telles équations.

### 1.1 SCHÉMAS NUMÉRIQUES POUR ÉQUATIONS DIFFÉRENTIELLES ORDINAIRES

L'une des utilisations importantes est la résolution numérique des équations différentielles ordinaires. La construction des schémas numériques conduit parfois à des méthodes implicites, c'est-à-dire nécessitant la résolution d'équations ordinaires. Ces méthodes peuvent présenter des avantages numériques même si elles peuvent parfois sembler plus compliquées. Pour s'en convaincre, considérons le problème simple

$$u'(t) = -u(t) \quad t \in ]0, T[, \quad \text{avec } u(0) = u_0,$$

dont la solution est donnée par  $u(t) = u_0 e^{-t}$ .

En considérant un pas de temps constant  $\Delta t > 0$ , le schémas d'Euler explicite appliqué à la résolution de cette équation conduit à la construction de la suite  $u^e = (u_n^e)_{n \in \mathbb{N}}$  définie par

$$u_{n+1}^e = (1 - \Delta t)u_n^e, \quad n \geq 0, \quad (4.2)$$

alors que le schéma d'Euler implicite conduirait à la suite  $u^i = (u_n^i)_{n \in \mathbb{N}}$

$$u_{n+1}^i = u_n^i - \Delta t u_{n+1}^i, \quad n \geq 0.$$

Pour  $u_n^i$  donné, la détermination de  $u_{n+1}^i$  se ramène ici à la résolution de l'équation  $x = u_n^i - x \Delta t$  dont la solution est évidente car l'équation à résoudre est linéaires... Nous avons donc

$$u_{n+1}^i = \frac{1}{1 + \Delta t} u_n^i, \quad n \geq 0. \quad (4.3)$$

## Comparaison explicite, implicite

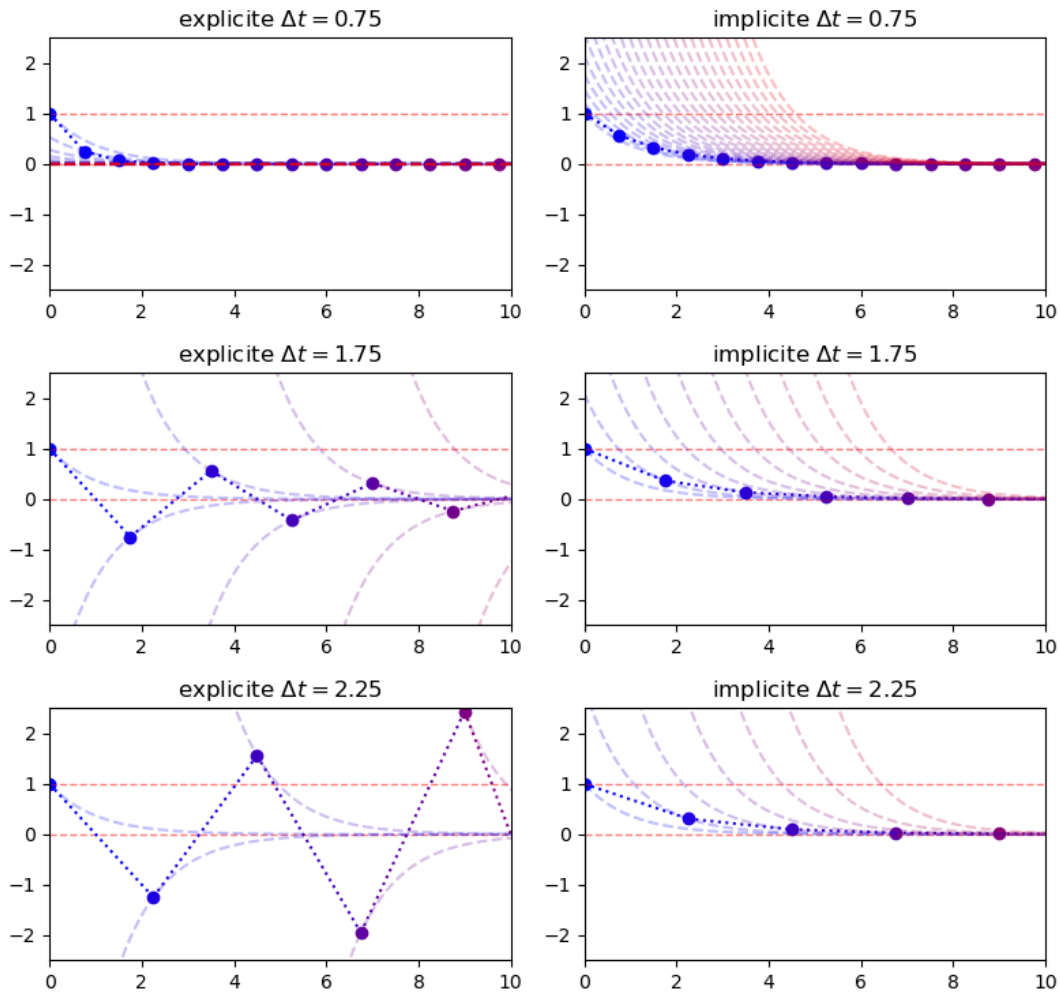


FIGURE 4.1 – Comparaison des méthodes explicites et implicites

La figure 4.1 montre les premiers termes de la suite obtenues par les méthodes explicites et implicites avec différentes valeurs du pas de temps  $\Delta t$ .

Pour la méthode explicite, on montre que, si l'on prend  $u_0 > 0$ ,

- ▷ si  $0 < \Delta t \leq 1$ , la suite est décroissante et converge vers 0, ce qui est le bon comportement ;
- ▷ si  $1 < \Delta t < 2$ , la suite prend alternativement des valeurs positives et négatives mais converge vers 0 ;
- ▷ si  $\Delta > 2$ , la suite diverge en prenant alternativement des valeurs positives et négatives de plus en plus grandes en valeur absolue.

Pour la méthode implicite, on montre que, si l'on prend  $u_0 > 0$ , la suite reste positive et tend vers 0 quelque soit la valeur du pas de temps  $\Delta t$ . Ainsi, le comportement de la solution approchée reste toujours cohérent avec celui de la solution exacte. La méthode implicite semble plus intéressante.

Cependant il n'est pas toujours possible de résoudre directement le problème posé par les schémas implicites. En particulier, lorsque la fonction n'est pas linéaire, la résolution n'admet pas toujours une seule solution et les solutions n'ont pas toujours d'expression analytique. Voici un exemple

illustrant ce propos.

$$u'(t) = e^{u(t)} \quad t \in ]0, T[, \quad \text{avec} \quad u(0) = u_0. \quad (4.4)$$

Ici le schéma d'Euler implicite s'écrit

$$u_{n+1} = u_n + \Delta t e^{u_{n+1}}, \quad n \geq 0.$$

Pour  $n \geq 0$ , cette équation ne pourra être résoluble que de manière approchée (numériquement). Sachant que cette résolution devra être faite à chaque itération, il faudra qu'elle soit efficace, c'est-à-dire rapide (peu coûteuse en temps de calcul et taille mémoire) et précise (afin de ne pas accroître l'erreur locale de la méthode à un pas implicite).

---

## 1.2 MÉTHODE DE TIR POUR LES PROBLÈMES AUX LIMITES DU SECOND ORDRE

---

Dans un problème de tir à canon, on est souvent amené à déterminer l'angle vertical d'orientation du canon afin d'atteindre une cible précise.

En une dimension d'espace, le problème peut se mettre sous la forme (4.5) où  $f$  est une certaine fonction donnée de manière implicite sous la forme suivante.

Etant donnés deux réels  $x_0$  et  $x_T$  donnés, on cherche une solution du problème

$$\begin{cases} x''(t) = f(t, x(t)), & t \in ]0, T[, \\ x(0) = x_0, \\ x(T) = x_T. \end{cases} \quad (4.5)$$

On obtient ici un problème aux limites et non un problème de Cauchy pour une équation différentielle ! Si l'on dispose d'un bon solveur d'EDO, on peut en faire usage. On peut en effet transformer ce problème en un problème de Cauchy (en une edo), en introduisant une inconnue nécessaire pour définir une condition initiale.

En effet, si pour  $v \in \mathbb{R}$  on considère le problème

$$\begin{cases} z''(t) = f(t, z(t)), & t \in ]0, T[, \\ z(0) = x_0, \\ z'(0) = v, \end{cases} \quad (4.6)$$

on a alors un problème de Cauchy pour une EDO. Si l'on pose alors  $g(t, v)$  la solution de cette EDO, en l'évaluant à  $t = T$  on définit une fonction  $g(T, v)$  de la seule variable  $v$ .

Le problème de détermination de la vitesse initiale revient alors à résoudre l'équation d'inconnue  $v : G(v) = 0$ , avec  $G(v) = g(T, v) - x_T$ .

Il apparaît donc que dans certaines équations non linéaires, l'expression de la fonction dont on cherche une racine peut être compliquée : ici une évaluation de la fonction  $G$  nécessite une résolution d'équation différentielle. Il est donc nécessaire de chercher des méthodes numériques de résolution d'équations non-linéaires qui soient efficaces et si possible sans recours au calcul des dérivées.

---

## 2 POSITION CORRECTE DU PROBLÈME

---

Avant de nous lancer dans la construction des schémas numériques pour le problème 4.1 d'équations non-linéaires, il est important de répondre aux questions suivantes.

1. Le problème admet-il une solution ?
2. Si oui cette solution est-elle unique ?
3. Cette solution dépend-t-elle continûment des données? Autrement dit est-elle stable vis-à-vis des données du problème ?
4. La solution du problème si elle existe a-t-elle une régularité particulière ?

La réponse à ces questions est ce que l'on appelle vérification du côté *bien posé* du problème. Elle est nécessaire lorsqu'on souhaite résoudre numériquement (de manière approchée) tout problème donné. Si le point 3 est négligé, les conséquences peuvent être dramatiques au niveau numérique.

## 2.1 EXISTENCE DE SOLUTION

Puisqu'on est dans un cadre scalaire (c'est-à-dire  $f : \mathbb{R} \mapsto \mathbb{R}$ ), l'existence de solution est une simple conséquence du théorème de Rolle (Thm. 1.4).

**PROPOSITION 4.2** – cas où l'équation est sous la forme  $f(x) = 0$

Soit  $I = [a, b]$  et  $f : I \rightarrow \mathbb{R}$  continue telle que  $f(a)f(b) < 0$ , alors il existe  $x^* \in I$  tel que  $f(x^*) = 0$ .

**PROPOSITION 4.3** – cas où l'équation est sous la forme  $x = g(x)$

Soit  $I = [a, b]$  et  $g : I \rightarrow \mathbb{R}$  continue telle que  $g(I) \subset I$ , alors il existe  $x^* \in I$  tel que  $g(x^*) = x^*$ .

*Démonstration.* On applique le résultat précédent à  $f(x) = g(x) - x$  ◻

**REMARQUE 4.4** – accessibilité de l'intervalle  $I$

Les résultats que nous venons de présenter, font l'hypothèse que l'intervalle  $I$  est connu. Mais ce n'est pas toujours le cas en pratique. Ainsi,  $I$  est lui même inconnu et est appelé *domaine d'attraction* de la solution dans le cas d'un problème de point fixe.

Lorsque l'équation est sous la forme  $f(x) = 0$ , un procédé constructif permet aussi de définir l'intervalle  $I$  : on se fixe un point  $a$  et un pas  $\Delta$ . Puis on se déplace selon ce pas de part et d'autre de  $a$ , jusqu'au premier changement de signe de  $f$ . On définit alors l'intervalle  $I$ . Pour  $\Delta$  suffisamment petit, cette procédure approche directement la racine de l'équation et on appelle la schéma ainsi construit *méthode de recherche incrémentale*.

On retient alors à ce stade que certaines preuves de la position correcte des équations non-linéaires cachent en elles mêmes des schémas numériques, certes basiques, mais exploitables pour la résolution de ces équations.

## 2.2 NOTION DE CONDITIONNEMENT

Intéressons nous à présent à la stabilité. C'est-à-dire à l'effet, sur la solution, des perturbations sur les données.



$m$	$\bar{x}^*$	$ \bar{x}^* - x^* $	conditionnement
1	1.000000010000000	1.000E-08	1.000E+00
2	0.999900000000000	1.000E-04	1.000E+04
3	1.002154434690032	2.154E-03	2.154E+05
4	0.990000000000000	1.000E-02	1.000E+06
5	1.025118864315096	2.512E-02	2.512E+06

TABLE 4.1 – Influence du conditionnement sur la recherche de zéro

**DÉFINITION 4.5 – Conditionnement**

On appelle *nombre de conditionnement* d'un algorithme, le facteur d'amplification (ou de réduction) de l'erreur d'évaluation d'un algorithme.

Si ce facteur s'intéresse aux erreurs relatives on parle de *conditionnement relatif* ou simplement *conditionnement*. Si par contre ce facteur ne s'intéresse qu'aux erreurs absolues, on parle de *conditionnement absolu*.

Le problème sera dit *bien conditionné* si le conditionnement est petit, c'est-à-dire proche de 1. Il sera dit *mal conditionné* si ce nombre est très grand par rapport à 1.

**PROPOSITION 4.6 – conditionnement absolu**

Supposons  $f$  de classe  $\mathcal{C}^p$ ,  $p \geq 0$  et soient  $x^*$  et  $\bar{x}^*$  solutions de

$$f(x^*) = 0, \quad f(\bar{x}^*) + \eta(\bar{x}^*) = 0,$$

où  $\eta$  est une perturbation (erreur de mesure) de  $f$  supposée bornée au voisinage de  $x^*$ .

Si  $x^*$  est une racine de multiplicité  $m < p$  de  $f$  alors

$$|\bar{x}^* - x^*| \lesssim m! \left| \frac{\eta(\bar{x}^*)}{f^{(m)}(x^*)} \right|^{1/m}, \quad (4.7)$$

*Démonstration.* Effectuons le développement limité de la fonction  $f$  autour du point  $x^*$  avec un reste de Lagrange. Nous avons

$$f(\bar{x}^*) = f(x^*) + \frac{1}{m!}(\bar{x}^* - x^*)^m f^{(m)}(\xi),$$

où  $\xi$  est dans l'intervalle  $[\bar{x}^*, x^*]$  ou  $[x^*, \bar{x}^*]$ . Nous en déduisons que

$$-\eta(\bar{x}^*) = \frac{1}{m!}(\bar{x}^* - x^*)^m f^{(m)}(\xi).$$

Par continuité, nous obtenons la majoration du théorème

$$|\bar{x}^* - x^*| \lesssim m! \left| \frac{\eta(\bar{x}^*)}{f^{(m)}(x^*)} \right|^{1/m},$$

ce qui termine la preuve. Notons que, par définition de la multiplicité, le terme  $f^{(m)}(x^*)$  n'est pas nul. ◻

Ainsi, le *conditionnement absolu* est donné par  $\kappa(f, x^*) = m! |f^{(m)}(x^*)|^{-1/m} |\eta(\tilde{x})|^{1/m-1}$ . Ce qui montre que la résolution des équations non-linéaires à racines multiples sera un problème moins bien conditionné (et même mal conditionné) que celui de la recherche de racines simples.

Prenons pour exemple la recherche du zéro des fonctions  $f_m : x \mapsto (x - 1)^m$  en supposant que l'erreur  $\eta$  est constante ( $\eta(x) = -10^{-8}$  la valeur de  $\eta$  est choisie négative pour que l'équation ait toujours des solutions). Le tableau 4.1 récapitule les résultats obtenus pour différentes valeurs de la multiplicité  $m$ .

Ainsi, si l'on détermine le zéro avec 8 chiffres significatifs (précision du calcul de  $10^{-8}$ ), le conditionnement pour  $m = 1$  permet de conserver la même erreur sur la solution, tandis que pour  $m = 4$  par exemple, l'erreur sur la solution est de  $10^{-2}$ , c'est-à-dire que l'erreur est amplifiée d'un facteur  $10^6$ . Il faut avoir conscience de ce phénomène d'amplification des erreurs lorsque l'on fait du calcul scientifique.

### 2.3 VITESSE DE CONVERGENCE - ORDRE DE CONVERGENCE

Comme nous le verrons, la recherche de solution approchée conduira à la construction de suites qui convergent vers la solution du problème. Il est donc nécessaire de quantifier cette convergence. Pour cela un rappel sur l'ordre de convergence des suites est nécessaire.

#### DÉFINITION 4.7 – ordre de convergence

Soit  $(x_n)_{n \in \mathbb{N}}$  une suite qui converge vers une limite  $x^*$ . La convergence de la suite  $(x_n)$  vers  $x^*$  est dite *d'ordre au moins  $p$* ,  $p \in [1, +\infty[$ , si

$$\exists n_0 \in \mathbb{N} : \forall n > n_0 \quad |x_{n+1} - x^*| \leq C|x_n - x^*|^p,$$

où  $C$  est une constante strictement positive et  $C < 1$  si  $p = 1$ . La convergence est dite *d'ordre exactement  $p$*  si on a de plus

$$\exists n_0 \in \mathbb{N} : \forall n > n_0 \quad c|x_n - x^*|^p \leq |x_{n+1} - x^*| \leq C|x_n - x^*|^p,$$

où  $c$  est une autre constante.

#### PROPOSITION 4.8 – critère d'ordre de convergence

Soit  $(x_n)_{n \in \mathbb{N}}$  une suite qui converge vers une limite  $x^*$  sans jamais être égale à  $x^*$ . Si la limite

$$\lim_{n \rightarrow \infty} \frac{|x_{n+1} - x^*|}{|x_n - x^*|^p} = K$$

existe pour un certain  $p \in [1, +\infty[$ , alors

- ▷ si  $p = 1$  et  $0 < K < 1$ , la suite converge linéairement (exactement d'ordre 1),
- ▷ si  $p > 1$  et  $0 < K$ , la suite converge à l'ordre exactement  $p$ .

La limite  $K$  est alors appelé *la constante asymptotique de l'erreur*.

Pour bien comprendre cette notion d'ordre de convergence, supposons que nous avons une suite  $(x_n)_{n \in \mathbb{N}}$  qui converge à l'ordre  $p$  vers  $x^*$  et que de plus nous avons

$$\lim_{n \rightarrow \infty} \frac{|x_{n+1} - x^*|}{|x_n - x^*|^p} = K > 0.$$

Le nombre  $d_n = -\log_{10}|x_n - x^*|$  mesure le nombre de chiffres (en base 10) de  $x_n$  et de  $x^*$  qui coïncident (à une constante additive près ne dépendant pas de  $n$ ). On peut écrire

$$d_{n+1} \approx p d_n - \log_{10}(K).$$

Ce qui montre qu'à une constante additive près, le nombre de chiffres exactes est multiplié par  $p$  à chaque itération. En effet on a pour  $p > 1$

$$d_{n+1} + \frac{\log_{10}(K)}{1-p} = p \left( d_n + \frac{\log_{10}(K)}{1-p} \right).$$

On peut donc estimer le nombre d'itérations nécessaires pour gagner un chiffre exacte. En particulier, lorsque la convergence est linéaire, comme ce sera le cas pour la plupart des méthodes

que nous verrons, même celles d'ordre élevée lorsqu'elles seront confrontées à des situations particulières comme celles de recherche des racines multiples.

**PROPOSITION 4.9 – Utilisation de la constante asymptotique d'erreur**

Soit  $(x_n)_{n \in \mathbb{N}}$  une suite qui converge linéairement avec une constante asymptotique d'erreur égale à  $K$ . Alors le nombre d'itérations nécessaires pour gagner un chiffre exacte est le plus petit entier supérieur à  $-1/\log_{10}(K)$ .

*Démonstration.* la formule de récurrence du nombre de chiffres exactes par itération est donnée ici par  $d_{n+1} = d_n - \log_{10}(K)$ . Ainsi après  $m$  itérations à partir de l'itération  $n$ , on a  $d_{n+m} = d_n - m \log_{10}(K)$ . Par conséquent,  $d_{n+m} = d_n + 1$  si et seulement si  $m = -1/\log_{10}(K)$  (arrondi à l'entier supérieur).  $\blacksquare$

Quelques fois il est suffisant d'estimer l'ordre de convergence au moyen de comparaisons :

**DÉFINITION 4.10**

Soient  $(x_n)_{n \in \mathbb{N}}$ ,  $(y_n)_{n \in \mathbb{N}}$  deux suites qui convergent respectivement vers  $x^*$ ,  $y^*$ . On dit que  $(x_n)$  converge plus vite que  $(y_n)$  si  $\lim_{n \rightarrow \infty} \frac{x_n - x^*}{y_n - y^*} = 0$ .

Ainsi, si la suite  $(x_n)$  converge à l'ordre  $q$ , si la suite  $(y_n)$  converge à l'ordre  $p$  et si  $(x_n)$  converge plus rapidement que  $(y_n)$ , alors  $q \geq p$ . Par conséquent si l'on ne dispose que de l'ordre  $p$  de  $(y_n)$ , on pourra dire pour  $(x_n)$  convergeant plus vite que  $(y_n)$  qu'elle converge à l'ordre **au moins**  $p$ .

### 3 MÉTHODES DE TYPE ENCADREMENT

Nous commençons par présenter deux méthodes de recherche de 0 qui ne sont pas très rapides mais qui ont l'intérêt d'être très robustes. En particulier, elles fonctionnent même pour les racines multiples.

#### 3.1 MÉTHODE DE LA DICHOTOMIE

Soit  $f : [a, b] \rightarrow \mathbb{R}$  continue telle que  $f(a)f(b) < 0$ . D'après le théorème de Rolle,  $f$  admet une racine dans  $]a, b[$ . Cette racine est certainement dans l'une des moitiés de l'intervalle  $[a, b]$ . C'est-à-dire si l'on pose  $c = \frac{a+b}{2}$ . L'un des intervalles  $[a, c]$  ou  $[c, b]$  nous placera dans la configuration de départ. On prend celui là et on rejette l'autre. On répète le processus, ce qui génère une suite d'intervalles  $([a_n, b_n])$  emboîtés, dont la longueur tend vers 0 et qui vérifie pour tout  $n$ ,  $f(a_n)f(b_n) < 0$ . La méthode de dichotomie suit donc la procédure suivante :

- ▷ on pose  $[a_0, b_0] = [a, b]$
- ▷  $[a_n, b_n]$  étant connu, on pose  $c = \frac{a_n + b_n}{2}$  et on teste le signe de  $f(a_n)f(c)$ 
  - ▷ Si cette valeur est strictement négative, la racine de  $f$  est dans  $[a_n, c]$  on pose alors  $[a_{n+1}, b_{n+1}] = [a_n, c]$
  - ▷ Si cette valeur est strictement positive, alors  $f(c)f(b_n) < 0$ , on pose alors  $[a_{n+1}, b_{n+1}] = [c, b_n]$
  - ▷ Si elle est nulle alors  $c$  est la racine.

La figure 4.2 illustre le comportement de la méthode de la dichotomie.

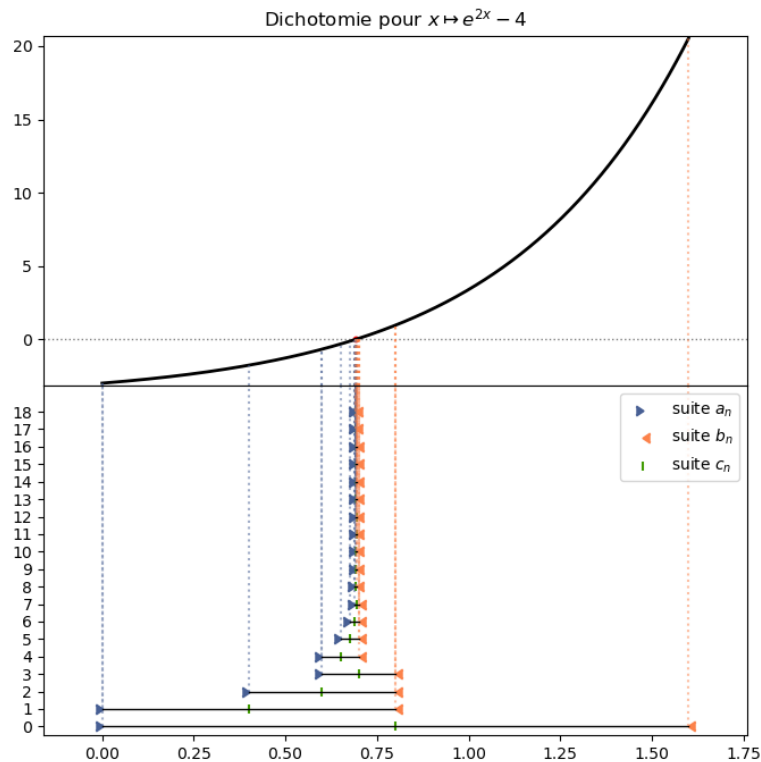


FIGURE 4.2 – Illustration de la méthode de la dichotomie

**PROPOSITION 4.11 – Propriétés de la méthode de la dichotomie**

Soit  $f$  continue sur  $[a, b]$  telle que  $f(a)f(b) < 0$  et qui possède un unique zéro dans  $]a, b[$  noté  $x^*$ . Soit  $(a_n), (b_n)$  les suites générées par la méthode de dichotomie, alors

- ▷  $x^* \in [a_n, b_n]$  pour tout  $n$  ;
- ▷  $[a_{n+1}, b_{n+1}] \subset [a_n, b_n]$  pour tout  $n$  ;
- ▷  $b_n - a_n = (b - a)/2^n$  ;
- ▷  $|a_n - x^*| \leq (b - a)/2^n$  et  $|b_n - x^*| \leq (b - a)/2^n$  ;
- ▷ les suites  $(a_n)$  et  $(b_n)$  convergent vers  $x^*$ .

‡ *Démonstration.* La preuve est triviale et laissée en exercice au lecteur. ⊙

Nous remarquons que, sous réserve que les hypothèses de la proposition précédente sont satisfaites, la méthode de la dichotomie converge toujours et il est possible d'estimer le nombre d'itérations nécessaires pour déterminer le zéro  $x^*$  avec une précision  $\epsilon$  choisie. En effet, à l'étape  $n$ , on est assuré que  $x^*$  se trouve dans l'intervalle  $[a_n, b_n]$  de longueur  $(b - a)/2^n$ . Ainsi, pour  $\epsilon$  fixé, le choix de

$$n = \mathbb{E} \left( \log_2 \left( \frac{b - a}{2\epsilon} \right) \right) + 1 = \mathbb{E} \left( \log_2 \left( \frac{b - a}{\epsilon} \right) \right)$$

permet d'assurer que  $(a_n + b_n)/2$  est proche de  $x^*$  à  $\epsilon$  près.

## ALGORITHME 4.1 – Dichotomie

```

def dichotomie(f, a, b, epsilon, verbose=False):
    an, bn = (a, b) if a < b else (b, a)
    fan, fbn = f(an), f(bn)
    la, lb, lc = [an], [bn], []
    if fan * fbn > 0:
        raise ValueError(f"Probleme dans dichotomie : {a}, {b}")
    while bn - an > epsilon:
        cn = .5*(an+bn)
        fcn = f(cn)
        if fan*fcn <= 0:
            bn, fbn = cn, fcn
        if fbn*fcn <= 0:
            an, fan = cn, fcn
        la.append(an)
        lb.append(bn)
        lc.append(cn)
    cn = .5*(an+bn)
    lc.append(cn)
    if verbose:
        return cn, np.asarray(la), np.asarray(lb), np.asarray(lc)
    return cn

```

La fonction `dichotomie` prend cinq arguments : la fonction `f` dont on cherche un zéro, deux réels `a` et `b` qui définissent l'intervalle de recherche, un réel `epsilon` qui est un petit paramètre utilisé pour arrêter l'algorithme lorsque la précision est atteinte et un booléen `verbose` qui permet de modifier la sortie (seulement la solution trouvée ou bien toutes les valeurs intermédiaires calculées).

Résumé de la méthode de dichotomie :

- ▷ *avantages* :
  - ▷ méthode simple à implémenter,
  - ▷ convergence certaine
- ▷ *inconvenients* :
  - ▷ hypothèse de départ contraignante (encadrement du zéro),
  - ▷ convergence lente.

### 3.2 MÉTHODE DE LA SÉCANTE

On fera attention au vocabulaire : la méthode de la sécante est appelée *method of false position* or *regula falsi* dans les publications anglophones tandis que la méthode de la fausse position est appelée *secant method*.

Cette méthode suit le même principe que la méthode de dichotomie avec pour seule différence qu'au lieu de prendre  $c_n$  comme le milieu de  $a_n$  et de  $b_n$ , on le prendra de manière équivalente comme la racine du polynôme interpolateur de Lagrange associé au noeuds  $a_n$  et  $b_n$ , ou comme l'abscisse du point d'intersection avec l'axe des abscisses de la droite passant par  $(a_n, f(a_n))$ , et  $(b_n, f(b_n))$ . C'est-à-dire

$$c_n = \frac{a_n f(b_n) - b_n f(a_n)}{f(b_n) - f(a_n)}.$$

Comme  $f(a_n)$  et  $f(b_n)$  sont toujours de signe différent (par définition de l'algorithme), le dénominateur ne s'annule jamais et la suite est bien définie.

On construit donc les trois suites  $(a_n)$ ,  $(b_n)$  et  $(c_n)$  par  $a_0 = a$ ,  $b_0 = b$ , et pour  $n \in \mathbb{N}$ ,

$$c_n = \frac{a_n f(b_n) - b_n f(a_n)}{f(b_n) - f(a_n)}, \quad a_{n+1} = \begin{cases} a_n & \text{si } f(a_n)f(c_n) < 0, \\ c_n & \text{si } f(a_n)f(c_n) > 0, \end{cases} \quad b_{n+1} = \begin{cases} c_n & \text{si } f(a_n)f(c_n) < 0, \\ b_n & \text{si } f(a_n)f(c_n) > 0, \end{cases} \quad (4.8)$$

Si  $f(c_n) = 0$  pour un certain  $n \in \mathbb{N}$ , on ne continue pas l'algorithme et les suites sont finies.

La figure 4.3 illustre le comportement de la méthode de la sécante.

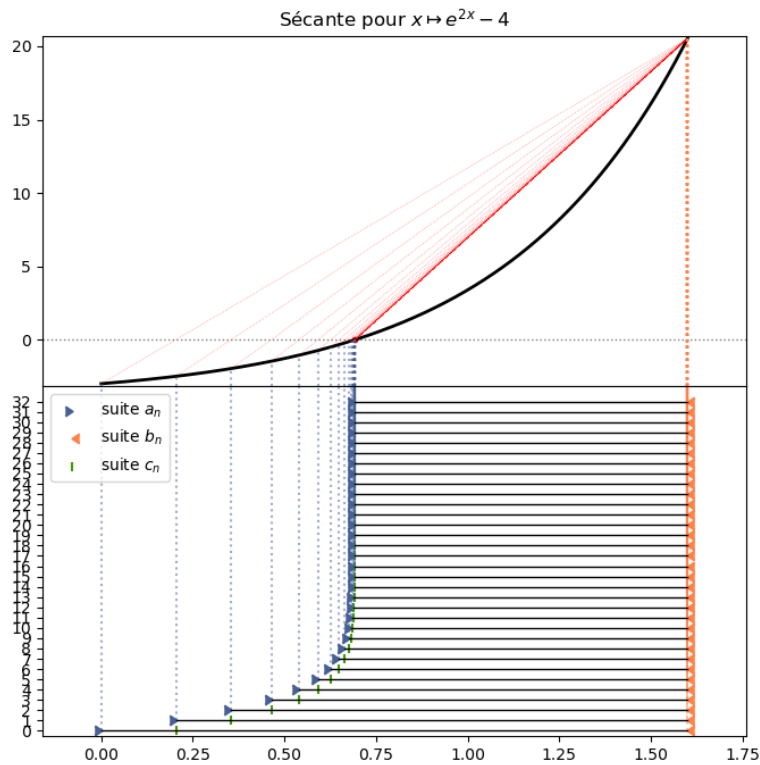


FIGURE 4.3 – Illustration de la méthode de la sécante

#### PROPOSITION 4.12 – Propriétés de la méthode de la sécante

Soit  $f$  continue sur  $[a, b]$  telle que  $f(a)f(b) < 0$  et qui possède un unique zéro dans  $]a, b[$  noté  $x^*$ . Soit  $(a_n)$ ,  $(b_n)$  et  $(c_n)$  les suites générées par la méthode de la sécante, alors

- ▷  $x^* \in [a_n, b_n]$  pour tout  $n$  ;
- ▷  $[a_{n+1}, b_{n+1}] \subset [a_n, b_n]$  pour tout  $n$  ;
- ▷ la suite  $(c_n)$  converge vers  $x^*$  ;

*Démonstration.* Pour fixer les idées, on suppose comme à la figure 4.3 que  $f(a) < 0$  et  $f(b) > 0$ . Supposons que les trois suites sont infinies, c'est-à-dire que  $f(c_n) \neq 0$  pour tout  $n \in \mathbb{N}$ . La suite  $(a_n)$  est croissante majorée, elle converge donc vers  $\alpha$ , la suite  $(b_n)$  est décroissante minorée, elle converge donc vers  $\beta$ . Par continuité, on a  $f(\alpha) \leq 0$  et  $f(\beta) \geq 0$ . Deux cas se présentent alors.

Premier cas :  $f(\alpha) = f(\beta)$ . On a donc  $f(\alpha) = f(\beta) = 0$ . Cela prouve que  $\alpha = \beta = x^*$  et que la suite  $(c_n)$  tend vers  $x^*$ .

Deuxième cas :  $f(\alpha) \neq f(\beta)$ . Par continuité de  $f$ , on en déduit que la suite  $(c_n)$  converge vers  $\gamma$  avec

$$\gamma = \frac{\alpha f(\beta) - \beta f(\alpha)}{f(\beta) - f(\alpha)}.$$

Or à chaque étape  $c_n = a_n$  ou  $c_n = b_n$ . Supposons que  $c_n = a_n$  pour une infinité de  $n \in \mathbb{N}$ . Alors  $\gamma = \alpha$ . On en déduit que  $(\alpha - \beta)f(\alpha) = 0$ . Comme  $f(\alpha) \neq f(\beta)$ , on a  $\alpha \neq \beta$ . Donc  $f(\alpha) = 0$ , et donc  $\alpha = x^*$ , ce qui prouve bien que  $(c_n)$  tend vers  $x^*$ . De même si  $c_n = b_n$  pour une infinité de  $n \in \mathbb{N}$ , on montre que  $\beta = x^*$  et on obtient la même conclusion.  $\bullet$

On a donc prouvé que la suite  $(c_n)$  converge vers  $x^*$  mais on ne sait rien des deux suites  $(a_n)$  et  $(b_n)$ . On sait seulement qu'une des deux converge vers  $x^*$ . On peut également montrer que, si la fonction est strictement convexe ou strictement concave, l'une des deux suites  $(a_n)$  ou  $(b_n)$  reste constante.

#### PROPOSITION 4.13 – Convergence de la méthode de la sécante

Soit  $f \in C^2([a, b])$  telle que  $f(a)f(b) < 0$  et  $f''$  n'a aucune racine dans l'intervalle  $[a, b]$  (la fonction est strictement convexe ou strictement concave). Soit  $(a_n), (b_n)$  générées par la méthode de fausse position. Deux cas sont alors possibles :

▷ la suite  $(b_n)$  est constante, alors  $(a_n)$  converge linéairement vers la racine  $x^*$  de  $f$  et on a

$$K_1 = \lim_{n \rightarrow \infty} \frac{x^* - a_{n+1}}{x^* - a_n} = 1 + f'(x^*) \frac{x^* - b}{f(b)};$$

▷ la suite  $(a_n)$  est constante, alors  $(b_n)$  converge linéairement vers la racine  $x^*$  de  $f$  et on a

$$K_1 = \lim_{n \rightarrow \infty} \frac{x^* - b_{n+1}}{x^* - b_n} = 1 + f'(x^*) \frac{x^* - a}{f(a)}.$$

*Démonstration.* Pour fixer les idées, on suppose comme à la figure 4.3 que  $f(a) < 0$  et  $f(b) > 0$  et que  $f''(x) > 0$  pour  $x \in [a, b]$ . Les autres cas se traitent de manière identique. Comme la fonction  $f$  est strictement convexe, la courbe est toujours sous ses sécantes, c'est-à-dire que  $f(c_n) < 0$ , la suite  $(b_n)$  est donc constante.

Nous avons donc une seule suite à étudier, la suite  $(a_n)$  définie par

$$a_0 = a, \quad a_{n+1} = \phi(a_n) \quad \text{avec} \quad \phi(x) = \frac{xf(b) - bf(x)}{f(b) - f(x)}.$$

Comme  $f(x^*) = 0$ , nous avons  $\phi(x^*) = x^*$ . Nous avons donc

$$a_{n+1} - x^* = \phi(a_n) - \phi(x^*) = \phi'(\xi)(a_n - x^*),$$

où  $\xi$  est dans l'intervalle  $]a_n, x^*[$ . D'après la proposition précédente, la suite  $(a_n)$  converge vers  $x^*$  car elle est égale à la suite  $(c_n)$ . Nous avons donc

$$\lim_{n \rightarrow \infty} \frac{x^* - a_{n+1}}{x^* - a_n} = \phi'(x^*).$$

Dérivons la fonction  $\phi$ . Nous avons

$$\begin{aligned} \phi'(x) &= \frac{(f(b) - bf'(x))(f(b) - f(x)) + f'(x)(xf(b) - bf(x))}{(f(b) - f(x))^2} \\ &= \frac{f(b)(f(b) - f(x)) - (b - x)f'(x)}{(f(b) - f(x))^2}. \end{aligned}$$

Cette expression se simplifie pour  $x = x^*$  en

$$\phi'(x^*) = \frac{f(b) - (b - x^*)f'(x^*)}{f(b)},$$

ce qui termine la preuve.  $\bullet$

Cette proposition ne permet pas facilement d'estimer à l'avance le nombre d'itérations nécessaires pour que la méthode de la sécante donne un résultat  $c_n$  proche de  $x^*$  à  $\epsilon$  près pour  $\epsilon > 0$  fixé. En effet, la limite  $K_1$  ne peut pas être calculée puisqu'elle dépend de  $x^*$  inconnu.

Dans le cas de la figure 4.3 obtenue avec  $f(x) = e^{2x} - 4$ , nous avons  $x^* = \ln(2)$ . Ainsi, pour  $b = 1$ ,  $K_1 \simeq 0.2757$ . La convergence est donc plus rapide que celle obtenue pour la dichotomie. Pour  $b = 1.6$ ,  $K_1 \simeq 0.6467$ . La convergence est donc plus lente que celle obtenue pour la dichotomie.

#### ALGORITHME 4.2 – Sécante

```
def secante(f, a, b, epsilon, verbose=False):
    an, bn = (a, b) if a < b else (b, a)
    fan, fbn, fcn = f(an), f(bn), 2*epsilon
    la, lb, lc = [an], [bn], []
    if fan * fbn > 0:
        raise ValueError(f"Probleme dans secante: {a}, {b}")
    while abs(fcn) > epsilon:
        cn = (an*fbn - bn*fan) / (fbn - fan)
        fcn = f(cn)
        if fan*fcn <= 0:
            bn, fbn = cn, fcn
        if fbn*fcn <= 0:
            an, fan = cn, fcn
        la.append(an)
        lb.append(bn)
        lc.append(cn)
    cn = (an*fbn - bn*fan) / (fbn - fan)
    lc.append(cn)
    if verbose:
        return cn, np.asarray(la), np.asarray(lb), np.asarray(lc)
    return cn
```

La fonction `secante` prend cinq arguments : la fonction `f` dont on cherche un zéro, deux réels `a` et `b` qui définissent l'intervalle de recherche, un réel `epsilon` qui est un petit paramètre utilisé pour arrêter l'algorithme lorsque la précision est atteinte et un booléen `verbose` qui permet de modifier la sortie (seulement la solution trouvée ou bien toutes les valeurs intermédiaires calculées).

#### Résumé de la méthode de la sécante :

- ▷ *avantages* :
  - ▷ méthode simple à implémenter,
  - ▷ convergence certaine et souvent meilleure que pour la dichotomie ;
- ▷ *inconvenients* :
  - ▷ hypothèse de départ contraignante (encadrement du zéro),
  - ▷ complexité un peu supérieure à celle de la dichotomie,
  - ▷ convergence lente et très lente vers les racines multiples.

## 4 MÉTHODES DE TYPE INTERPOLATION

Nous présentons à présent deux méthodes de recherche de 0 qui sont plus efficaces. Le principe est de remplacer la fonction  $f$  par un polynôme interpolateur de bas degré qui "ressemble" à la fonction  $f$  lorsque l'on s'approche du 0 recherché.



## 4.1 MÉTHODE DE NEWTON

La méthode de Newton est certainement une des méthodes les plus utilisées pour chercher les zéros d'une fonction. Elle consiste à remplacer la fonction  $f$  dont on cherche le zéro par sa tangente (il faut donc que la fonction  $f$  soit dérivable).

Etant donné un point  $x_0$ , nous calculons une nouvelle approximation  $x_1$  en cherchant le zéro de la tangente à  $f$  au point  $x_0$ . Cette tangente est la droite  $T_f(x_0)$  d'équation

$$T_f(x_0) : y = f(x_0) + (x - x_0)f'(x_0).$$

Cette droite coupe l'axe des abscisses en  $x_1 = x_0 - f(x_0)/f'(x_0)$ . Cela permet de construire la suite  $(x_n)$  tant que la dérivée  $f'(x_n)$  ne s'annule pas :

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

Lorsque la fonction  $f$  a un domaine de définition qui n'est pas  $\mathbb{R}$ , il est possible qu'un terme  $x_n$  sorte du domaine de définition. La construction de la suite échoue, la méthode ne donne pas de résultat intéressant... La conclusion est qu'il est nécessaire de bien choisir le point de départ  $x_0$  de la suite pour assurer que la suite est constructible et qu'elle converge bien vers  $x^*$ .

La figure 4.4 illustre le comportement de la méthode de Newton dans un cas favorable.

**PROPOSITION 4.14 – Convergence de la méthode de Newton (racine simple)**

Soit  $f$  une fonction de classe  $\mathcal{C}^1$  sur  $[a, b]$  qui possède un unique zéro dans  $]a, b[$  noté  $x^*$  et qui vérifie  $f'(x^*) \neq 0$ . Alors il existe  $h > 0$  tel que, si  $x_0 \in [x^* - h, x^* + h]$ , alors la suite  $(x_n)$  définie par

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

est constructible et converge vers  $x^*$ .

Si de plus  $f$  est de classe  $\mathcal{C}^2$ , la convergence est quadratique : nous avons en effet

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

*Démonstration.* Supposons pour simplifier que  $f'(x^*) = \alpha > 0$ . L'autre cas se démontre de la même façon. Comme la fonction  $f'$  est continue autour de  $x^*$ ,

$$\forall \underline{c} < 1 < \bar{c}, \quad \exists h > 0 : |x - x^*| \leq h \implies \alpha \underline{c} \leq f'(x) \leq \alpha \bar{c}.$$

Le choix des constantes sera précisé plus loin. Nous avons donc

$$x_{n+1} - x^* = x_n - x^* - \frac{f(x_n)}{f'(x_n)}.$$

Or, un développement limité de  $f$  au point  $x_n$  donne

$$0 = f(x^*) = f(x_n) + (x^* - x_n)f'(\xi),$$

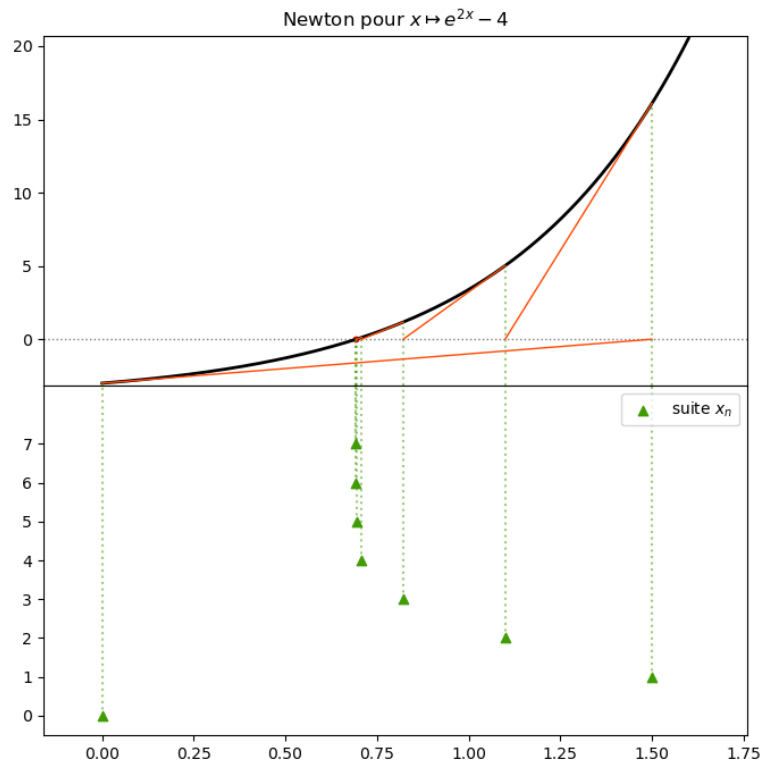


FIGURE 4.4 – Illustration de la méthode de Newton

pour  $\xi$  dans l'intervalle entre  $x^*$  et  $x_n$ . Nous obtenons donc

$$x_{n+1} - x^* = (x_n - x^*) \left( 1 - \frac{f'(\xi)}{f'(x_n)} \right)$$

Or

$$\left. \begin{array}{l} \alpha \underline{c} \leq f'(\xi) \leq \alpha \bar{c} \\ \alpha \underline{c} \leq f'(x_n) \leq \alpha \bar{c} \end{array} \right\} \implies \frac{\underline{c} - \bar{c}}{\underline{c}} \leq 1 - \frac{f'(\xi)}{f'(x_n)} \leq \frac{\bar{c} - \underline{c}}{\bar{c}} \implies \left| 1 - \frac{f'(\xi)}{f'(x_n)} \right| \leq \frac{\bar{c} - \underline{c}}{\underline{c}}.$$

Prenons par exemple  $\underline{c} = 9/10$  et  $\bar{c} = 11/10$ , nous concluons que

$$|x_{n+1} - x^*| \leq \frac{2}{9} |x_n - x^*|.$$

Ainsi, quitte à diminuer  $h$ , on peut supposer que tous les termes de la suite sont dans l'intervalle  $[x^* - h, x^* + h]$  et une récurrence immédiate donne  $|x_n - x^*| \leq (2/9)^n h$ , ce qui permet de conclure à la convergence de la suite.

Si de plus  $f \in \mathcal{C}^2([a, b])$  alors le développement limité autour du point  $x_n$  peut être prolongé en

$$0 = f(x^*) = f(x_n) + (x^* - x_n) f'(x_n) + \frac{1}{2} (x^* - x_n)^2 f''(\xi),$$

pour  $\xi$  dans l'intervalle entre  $x^*$  et  $x_n$ . On en déduit que

$$x_{n+1} - x^* = \frac{1}{2} (x_n - x^*)^2 \frac{f''(\xi)}{f'(x_n)},$$

ce qui permet de conclure après passage à la limite. ⊙

La propriété précédente assure la convergence de la méthode seulement lorsque le point de départ  $x_0$  est “suffisamment” proche du zéro recherché (avec des hypothèses de régularité). Il n'est pas possible d'améliorer dans le cas général ce résultat comme on peut le voir grâce à l'illustration proposée à la figure 4.5 :

- ▷ lorsque  $x_0$  est à gauche de la racine  $x^*$ , la concavité de la fonction assure que la suite  $(x_n)$  converge vers  $x^*$  ;
- ▷ lorsque  $x_0$  est proche de 0 ou de 1, la suite converge vers un 2-cycle, c'est-à-dire que les valeurs oscillent entre les deux valeurs ;
- ▷ lorsque  $x_0$  est dans la partie convexe ( $x_0 = 2$  par exemple), le comportement n'est pas prévisible : sur le dessin, la suite commence par osciller autour d'un 2-cycle puis converge brutalement vers  $x^*$ ...

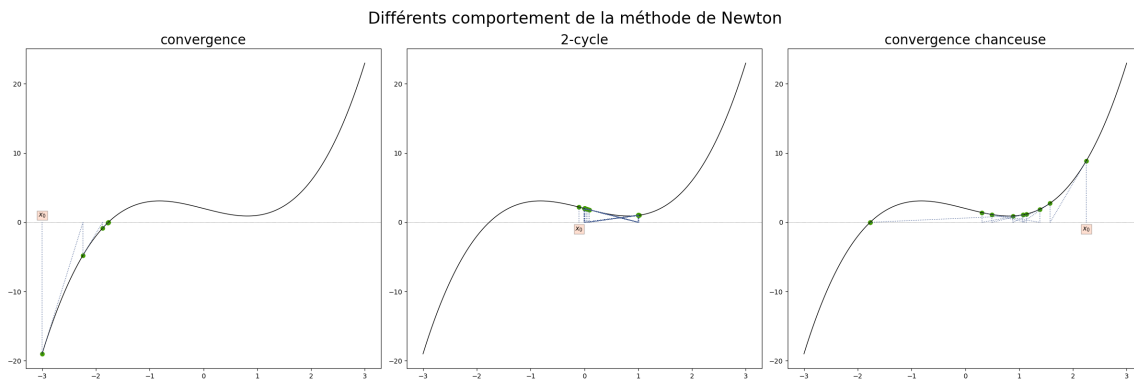


FIGURE 4.5 – Illustration de différents comportements de la méthode de Newton

#### ALGORITHME 4.3 – Newton

```
def newton(f, df, x0, epsilon, verbose=False):
    x = x0
    fx, dfx = f(x), df(x)
    lx = [x]
    n, Nmax = 0, 100
    while abs(fx) > epsilon * min(1, abs(dfx)) and n < Nmax:
        n += 1
        x -= fx / dfx
        fx, dfx = f(x), df(x)
        lx.append(x)
    if verbose:
        return x, np.asarray(lx)
    return x
```

La fonction `newton` prend cinq arguments : la fonction `f` dont on cherche un zéro, sa dérivée `df`, un réel `x0` qui sera le premier terme de la suite, un réel `epsilon` qui est un petit paramètre utilisé pour arrêter l'algorithme lorsque la précision est atteinte et un booléen `verbose` qui permet de modifier la sortie (seulement la solution trouvée ou bien toutes les valeurs intermédiaires calculées).

#### Résumé de la méthode de Newton :

- ▷ *avantages* :
  - ▷ méthode extrêmement rapide (toujours plus rapide que la sécante) ;

- ▷ hypothèse moins contraignante car il n'est pas nécessaire d'avoir un encadrement du zéro ;
- ▷ *inconvenients* :
  - ▷ évaluation obligatoire de la dérivée ;
  - ▷ convergence non assurée et lente pour les racines multiples.

## 4.2 MÉTHODE DE LA FAUSSE POSITION

L'idée de la méthode de la fausse position est de remplacer  $f'(x_n)$  dans la méthode de Newton par autre chose car il est rare de connaître en pratique une expression analytique de  $f'$  et on ne peut en avoir qu'une approximation. Cela peut arriver par exemple lorsque la fonction  $f$  est le résultat d'un calcul informatique ou bien une reconstruction à partir de données expérimentales.

On remplace  $f'(x_n)$  dans la méthode de Newton par

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

La méthode s'écrit ainsi : on choisit  $x_0 = a$  et  $x_1 = b$  ou l'inverse et on pose pour  $n \in \mathbb{N}$

$$x_{n+1} = x_n - f(x_n) \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}. \quad (4.9)$$


C'est-à-dire que lorsque  $x_n$  et  $x_{n-1}$  sont construits, on construit  $x_{n+1}$  comme l'intersection de l'axe des abscisses et de la droite reliant les points  $f(x_n)$  et  $f(x_{n-1})$ .

La figure 4.6 illustre le comportement de la méthode de la fausse position dans un cas favorable.

Il n'y a pas toujours constructibilité de cette méthode. Sans hypothèses supplémentaires, il se peut que l'on sorte de l'intervalle de définition de la fonction ou bien que le dénominateur soit nul. On a toutefois un résultat de convergence locale.

### PROPOSITION 4.15 – Convergence de la méthode de la fausse position

On suppose que  $f$  est de classe  $\mathcal{C}^2$  sur  $[a, b]$ , qu'il existe un unique zéro de  $f$  dans  $[a, b]$ , noté  $x^*$ , et que  $f'(x^*) \neq 0$ . Il existe alors  $\epsilon > 0$  tel que pour tout  $x_0, x_1$  tels que  $|x_0 - x^*| < \epsilon$  et  $|x_1 - x^*| < \epsilon$ , la suite  $(x_n)$  converge vers  $x^*$  et l'erreur  $|x_n - x^*|$  est majorée par une suite qui converge vers 0 à l'ordre au moins  $q = (\sqrt{5} + 1)/2$ .

! *Démonstration.* La démonstration utilise des outils élémentaires d'analyse mais est un peu technique. Le lecteur intéressé pourra la trouver dans [5]. 

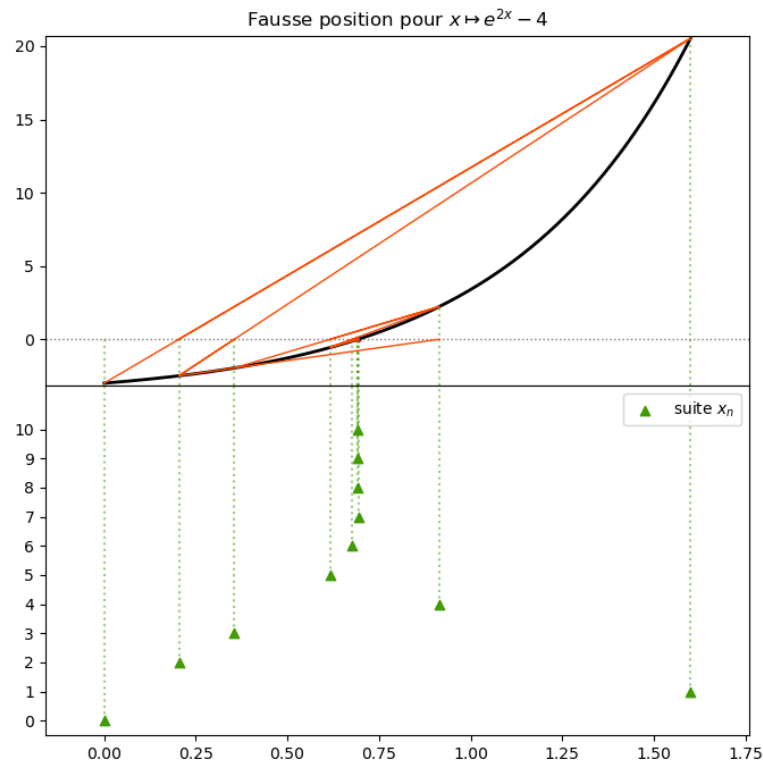


FIGURE 4.6 – Illustration de la méthode de la fausse position

## ALGORITHME 4.4 – Fausse position

```

def fausse_position(f, x0, x1, epsilon, verbose=False):
    x, xold = x1, x0
    fx, fxold = f(x), f(xold)
    dfx = (fx - fxold) / (x - xold)
    lx = [xold, x]
    n, Nmax = 0, 100
    while abs(fx) > epsilon * min(1, abs(dfx)) and n < Nmax:
        n += 1
        xold = x
        x -= fx / dfx
        fxold, fx = fx, f(x)
        dfx = (fx - fxold) / (x - xold)
        lx.append(x)
    if verbose:
        return x, np.asarray(lx)
    return x

```

La fonction `fausse_position` prend cinq arguments : la fonction `f` dont on cherche un zéro, deux réels `x0` et `x1` qui seront les deux premiers termes de la suite, un réel `epsilon` qui est un petit paramètre utilisé pour arrêter l'algorithme lorsque la précision est atteinte et un booléen `verbose` qui permet de modifier la sortie (seulement la solution trouvée ou bien toutes les valeurs intermédiaires calculées).

Résumé de la méthode de la fausse position :

- ▷ *avantages* :
  - ▷ méthode extrêmement rapide (presque aussi rapide que la méthode de Newton) ;
  - ▷ hypothèse moins contraignante car il n'est pas nécessaire d'avoir un encadrement du zéro ;
  - ▷ pas d'évaluation de la dérivée ;
- ▷ *inconvénients* :
  - ▷ le calcul de la dérivée approchée peut entraîner des erreurs d'arrondis importantes ;
  - ▷ convergence non assurée et lente pour les racines multiples.

- [1] G. Allaire. *Analyse numérique et optimisation*. Edition de l'Ecole Polytechnique, 2002.
- [2] H. Brezis. *Analyse Fonctionnelle Théorie et Applications*. Masson, 1983.
- [3] P. Ciarlet. *Introduction à l'analyse numérique matricielle et à l'optimisation*. Masson, 1994.
- [4] M. Crouzeix and A. L. Mignot. *Analyse numérique des équations différentielles*. Masson, 1992.
- [5] J. Demailly. *Analyse Numérique et Équations différentielles*. EDP Sciences, 2006.
- [6] G. Duvaut. *Mécanique des milieux continus*. Masson, 1990.
- [7] D. Euvrard. *Résolution numérique des EDP*. Masson, 1990.
- [8] L. Evans. *Partial Differential Equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, 2002.
- [9] E. Godlewski and P.-A. Raviart. *Hyperbolic systems of conservation laws*. Ellipses, 1991.
- [10] L. Hörmander. *Lectures on Nonlinear Hyperbolic Differential Equations*. Springer, 1997.
- [11] F. Hubert and J. Hubbard. *Calcul scientifique de la théorie à la pratique*. Vuibert, 2006.
- [12] B. Lucquin. *Équations aux dérivées partielles et leurs approximations*. Ellipses, 2004.
- [13] A. L. Pourhiet. *Résolution numérique des EDP*. Impr. du sud, 1988.
- [14] P.-A. Raviart and J.-M. Thomas. *Introduction à l'analyse numérique des équations aux dérivées partielles*. Masson, 1983.
- [15] M. Renardy and R. Rogers. *An introduction to partial differential equations*. Springer, 1993.
- [16] R. Richtmyer and K. Morton. *Difference methods for initial value problems*. Wiley-Interscience, New-York, 1967.
- [17] L. Schwartz. *Méthodes mathématiques pour les sciences physiques*. Hermann, 1965.
- [18] L. Schwartz. *Théorie des distributions*. Hermann, 1966.
- [19] D. Serre. *Matrices : Theory and Applications*. Springer, 2002.
- [20] D. Serre. *Systems of conservation laws 1*. Cambridge University Press, 2003.