

Contents

1	CM1	2
1.1	Modèles Discrets	2
1.1.1	Equation Générale des Modèles Discrets	2
1.1.2	Modèle de Croissance Géométrique	2
1.1.3	Modèle de croissance logistique discret	4
1.2	Modèles Continus	4
1.2.1	Motivation pour les Modèles Continus	4
1.2.2	Modèle de Malthus	4
1.2.3	Modèle de Verhulst (ou Logistique)	5
1.3	Conclusion	6
2	CM2	7
2.1	Introduction: Notion de Champ de Vecteurs et EDO	7
2.1.1	Généralités et Définitions	7
2.2	Analyse Qualitative des Solutions d'EDO	7
2.3	Champ de Vecteurs: Définitions et Propriétés	7
2.3.1	Vecteur Tangent à une Courbe Paramétrée	7
2.3.2	Lien entre Solution d'EDO et Vecteurs Vitesse	8
2.3.3	Définition d'un Champ de Vecteurs	8
2.4	Visualisation des Champs de Vecteurs (Implémentation Python)	8
2.4.1	Principe	8
2.4.2	Utilisation de <code>quiver</code> de Python	8
2.4.3	Contrôle de la Longueur des Vecteurs	9
2.5	Application: Recherche Approchée de Solutions (Python)	9
2.5.1	Objectif	9
2.5.2	Méthode Python	9
2.6	Exploitation des Champs de Vecteurs pour Comprendre l'Allure des Solutions	10
3	CM3	11
3.1	Analyse de la convergence	11
3.1.1	Approches	11
3.1.2	Valeur approchée par itération	11
3.1.3	Interprétation pratique de la vitesse de convergence	12
3.1.4	Nombre d'itérations pour gagner un chiffre en convergence linéaire	13
3.1.5	Linéarisation pour estimation graphique de q	13
3.1.6	Procédure pour déterminer q graphiquement	13
3.2	Python code pour l'estimation de la convergence (exemple)	14
4	CM4	16
4.1	Introduction à l'interpolation	16
4.1.1	Définition	16
4.1.2	Motivations	16
4.1.3	Exemples d'interpolation	17

4.2	Polynôme interpolateur de Lagrange	19
4.2.1	Définitions et propriétés	19
4.2.2	Estimation de l'erreur d'interpolation	20
4.2.3	Implémentation avec Python	20
4.3	Construction des polynômes d'interpolation de Lagrange	20
4.3.1	Interpolation dans la base canonique (Vandermonde)	20
4.3.2	Evaluation efficace : Algorithme de Horner	21
4.3.3	Interpolation dans la base duale: Formule de Lagrange et points barycentriques	21
5	CM5	23
5.1	Introduction à l'interpolation polynomiale	23
5.2	Interpolation de Lagrange	23
5.2.1	Formule de Lagrange	23
5.2.2	Exemple	24
5.3	Erreur d'interpolation	24
5.3.1	Formule de l'erreur d'interpolation	24
5.3.2	Analyse de l'erreur	24
5.3.3	Convergence	24
5.4	Interpolation de Newton	25
5.4.1	Différences divisées	25
5.4.2	Algorithme de calcul des différences divisées	25
5.4.3	Évaluation du polynôme de Newton : Formule de Horner-Newton	25
5.5	Comparaison des méthodes et complexité	26
5.5.1	Complexité	26
5.5.2	Optimisation et ajout de points	26
5.6	Conclusion	26
6	CM6	27
6.1	Polynômes de Tchebychev	27
6.1.1	Définition	27
6.1.2	Expression trigonométrique	27
6.1.3	Propriétés	28
6.1.4	Application : Polynôme de meilleure approximation uniforme	28
6.1.5	Application à l'interpolation polynomiale	28
6.2	Intégration numérique	29
6.2.1	Motivation et concept général	29
6.2.2	Construction des formules de quadrature	29
6.2.3	Exemples	30
6.2.4	Estimation de l'erreur	30
7	CM7	32
7.1	Introduction	32
7.2	Formules de Quadrature Élémentaires	32
7.3	Formules de Quadrature Composites	32
7.3.1	Principe de construction	32
7.3.2	Visualisation	33
7.4	Méthodes de Quadrature Composites Classiques	33
7.4.1	Méthode des Rectangles	33
7.4.2	Méthode des Trapèzes	34
7.4.3	Méthode du Point Milieu	35
7.4.4	Méthode de Simpson	37
7.5	Comparaison des Méthodes	38
7.6	Conclusion	39
7.7	Exercices	39

8	CM8	40
8.1	Introduction aux Méthodes de Newton-Cotes	40
8.2	Exemples de Formules et Cas Particuliers	40
8.3	Théorème et Erreur d'Intégration	41
8.4	Construction de Formules de Quadrature à Points Non Équidistants : Formule de Gauss-Legendre	41
8.4.1	Méthode de résolution pratique (pour 2 points)	42
9	CM9	44
9.1	Solution approchée d'Équations Différentielles Ordinaires (EDO)	44
9.1.1	Motivations et Définitions	44
9.1.2	Exemples d'EDO	45
9.1.3	Problème de Cauchy et Existence/Unicité de la solution	47
9.1.4	Schémas Numériques à un pas	48
9.1.5	Exemple: Modèle Proie-Prédateur (Lotka-Volterra)	51
10	CM10	53
10.1	Schémas numériques pour Équations Différentielles Ordinaires	53
10.1.1	Le Problème de Cauchy	53
10.1.2	Schémas à un pas explicites	53
10.1.3	Exemples de schémas à un pas explicites	53
10.2	Étude de convergence pour les EDO	54
10.2.1	Convergence	54
10.2.2	Consistance et Ordre d'un schéma	54
10.2.3	Stabilité	56
10.2.4	Convergence des schémas à un pas explicite	58
11	CM11	60
11.1	Généralités et exemples	60
11.1.1	Rappels ((EO), $f(x)=0$)	60
11.1.2	Définitions	60
11.1.3	Exemples	61
11.2	Partition Correcte du Problème (EO)	61
11.2.1	Existence et unicité de la solution	62
11.2.2	Propriétés utiles	62
11.3	Construction de schémas pour (EO)	63
11.3.1	Méthode de dichotomie (cas $f(x) = 0$)	63
11.3.2	Méthode de la fausse position (Regula Falsi)	64
11.4	Algorithmes : Méthodes Itératives	66
11.4.1	Convergence	66
12	CM12	67
12.1	Méthode d'itération	67
12.1.1	Principe	67
12.1.2	Représentation graphique	67
12.1.3	Algorithme	68
12.1.4	Convergence	69
12.2	Méthode de Newton	70
12.2.1	Principe	70
12.2.2	Comment construire x_{n+1} à partir de x_n ?	70
12.2.3	Interprétation géométrique	71
12.2.4	Algorithme	71
12.2.5	Convergence	72

Chapter 1

CM1

1.1 Modèles Discrets

Les modèles discrets considèrent que les changements de population se produisent à intervalles de temps distincts.

1.1.1 Equation Générale des Modèles Discrets

Considérons $N(t)$ comme la population d'individus à l'instant t . L'équation générale d'un modèle discret est donnée par la variation de la population entre deux instants discrets t et $t + \Delta t$:

$$N(t + \Delta t) - N(t) = \text{nombre de naissances} - \text{nombre de décès} + \text{nombre d'immigrations} - \text{nombre d'émigrations}$$

En termes de taux, nous pouvons écrire:

$$N(t + \Delta t) - N(t) = n - m + i - e$$

où:

- n représente le nombre de naissances pendant l'intervalle Δt .
- m représente le nombre de décès pendant l'intervalle Δt .
- i représente le nombre d'immigrations pendant l'intervalle Δt .
- e représente le nombre d'émigrations pendant l'intervalle Δt .

1.1.2 Modèle de Croissance Géométrique

Le modèle de croissance géométrique est un modèle discret simple qui décrit la croissance d'une population dans des conditions idéales, où les ressources sont illimitées.

Hypothèses

- **Solde migratoire nul:** On suppose que le nombre d'immigrations est égal au nombre d'émigrations, donc $i - e = 0$.
- **Croissance proportionnelle à la taille de la population:** Le nombre de naissances est proportionnel à la taille de la population, avec un taux de natalité λ , et le nombre de décès est proportionnel à la taille de la population, avec un taux de mortalité μ . Ainsi, pendant l'intervalle Δt :
 - Nombre de naissances: $n = \lambda \Delta t N(t)$
 - Nombre de décès: $m = \mu \Delta t N(t)$

Équation et Solution

En posant $N_n = N(t_n)$ où $t_n = n\Delta t$, l'équation du modèle devient:

$$N_{n+1} - N_n = \lambda\Delta t N_n - \mu\Delta t N_n$$

Soit en posant $z = \lambda - \mu$, le taux de croissance:

$$N_{n+1} - N_n = z\Delta t N_n$$

$$N_{n+1} = N_n + z\Delta t N_n = (1 + z\Delta t)N_n$$

En définissant le taux de croissance par unité de temps $c = z\Delta t$, on a:

$$N_{n+1} = (1 + c)N_n$$

La solution de cette équation de récurrence, avec condition initiale N_0 (taille initiale de la population), est:

$$N_n = (1 + c)^n N_0 = (1 + z\Delta t)^n N_0$$

Visualisation

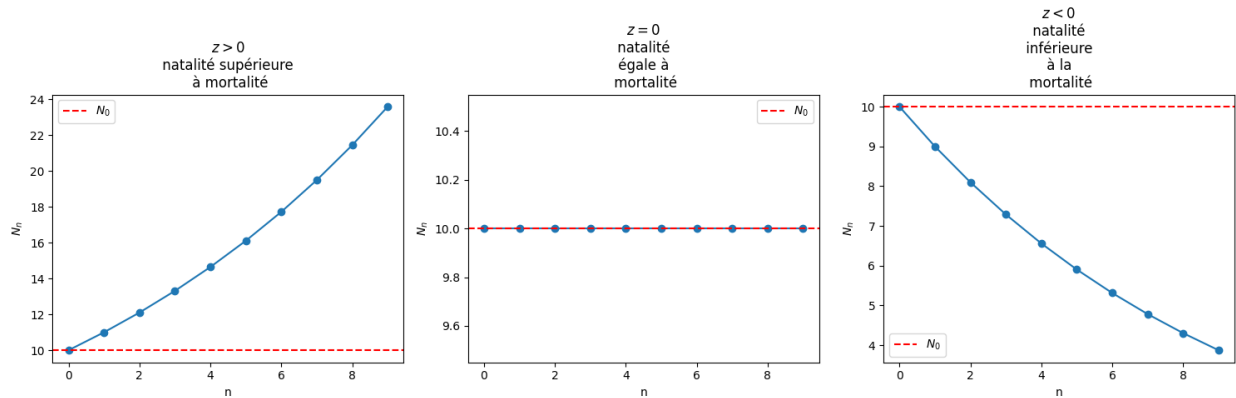


Figure 1.1: Visualisation du modèle de croissance géométrique pour différents taux de croissance z .

Propriétés du Modèle Géométrique

- **Tendance à l'infini pour $z > 0$:** Lorsque $z = \lambda - \mu > 0$, la population croît exponentiellement et tend vers l'infini lorsque $n \rightarrow \infty$. La solution $N(t) = N_0 e^{zt}$ est une approximation continue pour de petits Δt .
- **Croissance indéfinie pour $z > 0$:** Si $z > 0$, la population croît indéfiniment.
- **Extinction pour $z < 0$:** Si $z < 0$, la population décroît exponentiellement et tend vers l'extinction.
- **Population constante pour $z = 0$:** Si $z = 0$, la population reste constante au fil du temps, $N_n = N_0$ pour tout n .

Inconvénients du Modèle Géométrique

- **Croissance infinie irréaliste:** Une croissance infinie n'est pas réaliste dans le monde réel car les ressources sont limitées.
- **Approximation de partie entière:** Pour être rigoureux, on devrait écrire $E(\lambda\Delta t N_n)$ et $E(\mu\Delta t N_n)$ pour tenir compte du fait que le nombre d'individus doit être un entier, où $E(x)$ désigne la partie entière de x .

1.1.3 Modèle de croissance logistique discret

Le modèle de croissance logistique discret sera traité dans un exercice ultérieur.

1.2 Modèles Continus

Les modèles continus considèrent que les changements de population se produisent de manière continue dans le temps.

1.2.1 Motivation pour les Modèles Continus

Remark 1.2.1. L'utilisation de modèles continus est motivée par le fait que l'observation des populations sur des intervalles de temps très courts (Δt proche de 0) fournit beaucoup plus d'informations sur la dynamique de la population.

1.2.2 Modèle de Malthus

Le modèle de Malthus est le modèle continu le plus simple de croissance de population. Il est obtenu en passant à la limite du modèle géométrique lorsque $\Delta t \rightarrow 0$.

Hypothèses

- **Solde migratoire nul:** Comme pour le modèle géométrique, on suppose un solde migratoire nul.
- **Vitesses de natalité et de mortalité proportionnelles à la population:** On suppose que la vitesse de natalité et la vitesse de mortalité sont proportionnelles à la taille de la population à l'instant t .
 - Vitesse de natalité: $n(t) = \lambda N(t)$
 - Vitesse de mortalité: $m(t) = \mu N(t)$

Équation et Solution

Proposition 1.2.2. En reprenant l'équation de variation et en considérant les vitesses instantanées, l'équation différentielle du modèle de Malthus est:

$$N'(t) = \lim_{\Delta t \rightarrow 0} \frac{N(t + \Delta t) - N(t)}{\Delta t} = n(t) - m(t) = \lambda N(t) - \mu N(t)$$

Soit:

$$N'(t) = (\lambda - \mu)N(t)$$

En posant $z = \lambda - \mu$, on obtient:

$$N'(t) = zN(t)$$

Avec la condition initiale $N(0) = N_0$, la solution de cette équation différentielle est:

$$N(t) = N_0 e^{zt} = N_0 e^{(\lambda - \mu)t}$$

Propriétés du Modèle de Malthus

- Similaire au modèle géométrique en termes de comportement qualitatif, mais décrit la croissance de manière continue.
- **Croissance exponentielle pour $z > 0$:** Si $z = \lambda - \mu > 0$, la population croît exponentiellement.
- **Population constante pour $z = 0$:** Si $z = \lambda - \mu = 0$, la population reste constante.

- **Décroissance exponentielle pour $z < 0$:** Si $z = \lambda - \mu < 0$, la population décroît exponentiellement vers zéro.

Inconvénients du Modèle de Malthus

- **Croissance exponentielle irréaliste:** Comme le modèle géométrique, le modèle de Malthus prédit une croissance exponentielle infinie, ce qui n'est pas réaliste à long terme en raison de la limitation des ressources.
- **Ne prend pas en compte la limitation des ressources et l'interaction avec l'environnement.**

1.2.3 Modèle de Verhulst (ou Logistique)

Le modèle de Verhulst est une amélioration du modèle de Malthus qui prend en compte la limitation des ressources.

Idée Centrale

Definition 1.2.3. Limiter la croissance de la population à un seuil maximal K , appelé *capacité biotique* ou *capacité de charge* du milieu.

Hypothèses

- **Solde migratoire nul.**
- **Taux de natalité fonction affine décroissante de la population:** Le taux de natalité diminue à mesure que la population approche de la capacité biotique K . On le modélise par une fonction affine décroissante: $\lambda = \lambda_0(1 - \frac{N(t)}{K})$, où λ_0 est le taux de natalité maximal (lorsque $N(t)$ est très petit).
- **Taux de mortalité fonction affine croissante de la population:** Le taux de mortalité augmente à mesure que la population approche de K . On le modélise par une fonction affine croissante: $\mu = \mu_0(1 + \frac{N(t)}{K})$, où μ_0 est le taux de mortalité minimal (lorsque $N(t)$ est très petit). Pour simplifier, on prend souvent μ constant. Dans les notes, il est considéré comme une fonction affine croissante $\mu = -\mu_1(1 - \frac{N(t)}{K})$, ce qui implique que μ diminue quand $N(t)$ augmente, ce qui n'est pas biologiquement réaliste. On corrigera par $\mu = \mu_0 + \mu_1 \frac{N(t)}{K} = \mu_0(1 + \frac{N(t)}{K})$ ou simplement μ constant.

En utilisant la version simplifiée avec μ constant et en posant $\lambda = \lambda_0(1 - \frac{N(t)}{K})$, et $z_0 = \lambda_0 - \mu$, le taux de croissance intrinsèque maximal, on obtient:

$$z = \lambda - \mu = \lambda_0(1 - \frac{N(t)}{K}) - \mu = (\lambda_0 - \mu) - \frac{\lambda_0}{K}N(t) = z_0 - \frac{\lambda_0}{K}N(t)$$

On approche souvent $\lambda_0 \approx z_0$, et on pose simplement $z \approx z_0(1 - \frac{N(t)}{K})$.

Équation et Solution

Proposition 1.2.4. L'équation différentielle du modèle de Verhulst est alors:

$$N'(t) = zN(t) = z_0 \left(1 - \frac{N(t)}{K}\right) N(t)$$

Avec condition initiale $N(0) = N_0$. La solution de cette équation différentielle est donnée par:

$$N(t) = \frac{K}{1 + \left(\frac{K}{N_0} - 1\right) e^{-z_0 t}}$$

Visualisation

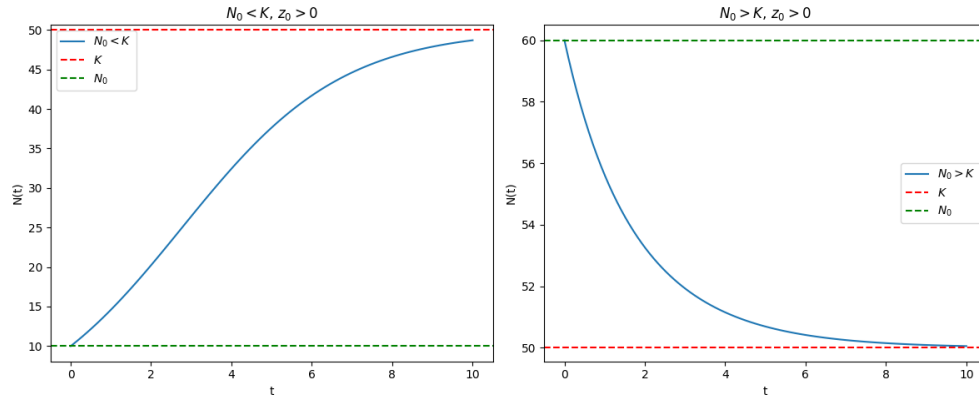


Figure 1.2: Visualisation du modèle de Verhulst pour différentes conditions initiales N_0 par rapport à la capacité biotique K .

1.3 Conclusion

Nous avons exploré les modèles discrets (géométrique) et continus (Malthus et Verhulst) pour la modélisation de populations. Le modèle géométrique et le modèle de Malthus, bien que simples, présentent des limitations importantes, notamment la prédiction d'une croissance infinie. Le modèle de Verhulst améliore ces modèles en introduisant la notion de capacité biotique, offrant une description plus réaliste de la dynamique des populations en tenant compte de la limitation des ressources.

Chapter 2

CM2

2.1 Introduction: Notion de Champ de Vecteurs et EDO

2.1.1 Généralités et Définitions

Nous allons étudier la notion de champ de vecteurs associé à une équation différentielle ordinaire (EDO).

Les modèles continus de la dynamique des populations sont des exemples de problèmes de Cauchy pour les EDOs.

Considérons une EDO du type :

$$y'(t) = f(t, y(t)), \quad t \in]t_0, T[, \quad (2.1)$$

avec la condition initiale :

$$y(t_0) = y_0, \quad (2.2)$$

où $y :]t_0, T[\rightarrow \mathbb{R}$ et $f :]t_0, T[\times \mathbb{R} \rightarrow \mathbb{R}$ est une fonction donnée, et $(t, x) \mapsto f(t, x)$.

2.2 Analyse Qualitative des Solutions d'EDO

Si l'on veut résoudre analytiquement une EDO, c'est-à-dire donner l'expression de $t \mapsto y(t)$, alors c'est terminé. Dans de nombreux cas, il suffit d'étudier la fonction $t \mapsto y(t)$.

Si l'on ne sait pas déterminer la solution analytique, on peut suivre une approche en deux étapes pour comprendre les solutions :

1. S'assurer de l'existence et de l'unicité de la solution, et de sa stabilité vis-à-vis des données du problème.
2. Puis analyser les propriétés qualitatives de cette solution par une simple analyse de $f(t, x)$. C'est ici qu'interviennent les champs de vecteurs.

2.3 Champ de Vecteurs: Définitions et Propriétés

2.3.1 Vecteur Tangent à une Courbe Paramétrée

Considérons une courbe paramétrée $t \mapsto \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} t \\ g(t) \end{pmatrix}$. Le vecteur tangent \vec{v} à cette courbe est donné par :

$$\begin{aligned} \vec{v} &= \begin{pmatrix} x'(t) \\ y'(t) \end{pmatrix} = \begin{pmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{pmatrix} = \begin{pmatrix} \frac{dx}{dy} \frac{dy}{dt} \\ \frac{dy}{dt} \end{pmatrix} = \frac{dx}{dt} \begin{pmatrix} 1 \\ \frac{dy}{dx} \end{pmatrix} \\ &= \frac{1}{\frac{dt}{dx}} \begin{pmatrix} 1 \\ g'(x) \end{pmatrix} = \frac{1}{\dot{x}(t)} \begin{pmatrix} 1 \\ \dot{y}(t) \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{\dot{y}(t)}{\dot{x}(t)} \end{pmatrix} = \begin{pmatrix} 1 \\ \dot{y}(t) \end{pmatrix} \end{aligned}$$

Si $x(t) = t$, alors $\dot{x}(t) = 1$, et le vecteur tangent devient $\vec{v} = \begin{pmatrix} 1 \\ \dot{y}(t) \end{pmatrix}$.

2.3.2 Lien entre Solution d'EDO et Vecteurs Vitesse

Proposition 2.3.1. y est solution de l'EDO $y'(t) = f(t, y(t))$ si et seulement si les vecteurs vitesses de la courbe paramétrée $t \mapsto \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} t \\ y(t) \end{pmatrix}$ au point t sont donnés par $u(t) = \begin{pmatrix} 1 \\ f(t, y(t)) \end{pmatrix}$.

2.3.3 Définition d'un Champ de Vecteurs

Définition 2.3.2. Soit $V : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ un champ de vecteurs, défini par $(t, y) \mapsto V(t, y)$. Si ce champ de vecteurs est associé à l'EDO $y'(t) = f(t, y(t))$, alors $V(t, y) = \begin{pmatrix} 1 \\ f(t, y) \end{pmatrix}$.

2.4 Visualisation des Champs de Vecteurs (Implémentation Python)

2.4.1 Principe

Pour dessiner un champ de vecteurs, à chaque point $P = (P_x, P_y)$, on trace le vecteur $V \in V(P) = (V_x, V_y)$. On choisit une constante positive pour représenter les vecteurs trop longs.

Figure 2.1: Schéma de principe pour le dessin d'un champ de vecteurs.

2.4.2 Utilisation de quiver de Python

Pour implémenter le dessin de champs de vecteurs en Python, on utilise la fonction `quiver` de `matplotlib.pyplot`. Les arguments principaux sont : `plt.quiver(Px, Py, Vx, Vy, angles='xy', scale)`.

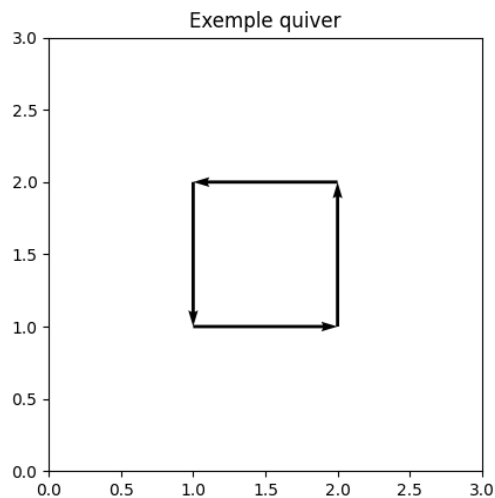


Figure 2.2: Exemple d'utilisation de quiver

2.4.3 Contrôle de la Longueur des Vecteurs

On peut ajouter un paramètre pour contrôler la longueur des vecteurs. Il est souvent nécessaire de normaliser les vecteurs pour une visualisation claire du champ de vecteurs.

Pour normaliser les vecteurs (V_x, V_y) , on calcule d'abord la norme :

$$\text{norm} = \sqrt{V_x^2 + V_y^2} \quad (2.3)$$

Puis on normalise chaque composante :

$$\begin{aligned} V_x &= V_x / \text{norm} \\ V_y &= V_y / \text{norm} \end{aligned}$$

2.5 Application: Recherche Approchée de Solutions (Python)

2.5.1 Objectif

On cherche une solution approchée de l'EDO $y'(t) = f(t, y(t))$, pour $t \in [t_0, t_0 + T]$ avec $y(t_0) = y_0$, en utilisant Python. Pour cela, il suffit de dessiner en quelques points où aboutit cette solution.

2.5.2 Méthode Python

Définissons un exemple de champ de vecteur avec $f(t, y) = r \cdot y \cdot (1 - y/k)$.

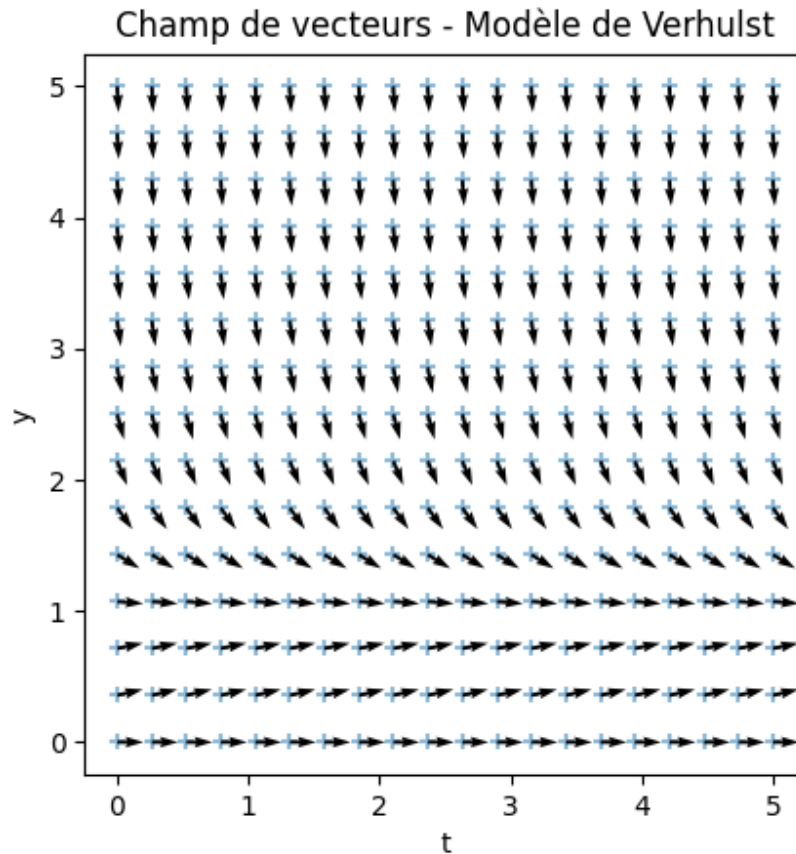


Figure 2.3: Champ de vecteurs pour le modèle de Verhulst.

2.6 Exploitation des Champs de Vecteurs pour Comprendre l'Allure des Solutions

Si l'on connaît les valeurs minimales et maximales de la solution, on peut avoir une idée de l'allure de la solution en observant le champ de vecteurs. Par exemple, si les vecteurs pointent vers le haut dans une certaine région, et vers le bas dans une autre, on peut déduire le comportement qualitatif des solutions dans ces régions.

Chapter 3

CM3

3.1 Analyse de la convergence

On va essayer de construire des polynômes qui passent par un ensemble (nuage) de points donnés.

Si ces points sont les valeurs d'une fonction, on aimerait :

- avoir un polynôme construit et d'autant plus proche de la fonction que le nombre de points est grand.

C'est-à-dire, est-ce que la suite des "meilleurs" polynômes tend vers la fonction lorsque le nombre de points tend vers l'infini?

Question : Comment quantifier cette convergence? C'est-à-dire à quelle vitesse (ordre) cette convergence a lieu.

3.1.1 Approches

- **Approche 1 :** Approximation linéaire
 - Moindre carré de degré 1
- **Approche 2 :** Polynôme d'ordre 1
 - Interpolation polynomiale (Lagrange)
- **Approche 3 :** Autres approches
 - Splines, ondelettes, etc.

3.1.2 Valeur approchée par itération

Définition de convergence

Definition 3.1.1. Soit $(x_n)_{n \in \mathbb{N}} \subset \mathbb{R}^d$ une suite qui converge vers $x^* \in \mathbb{R}^d$, pour une norme $\|\cdot\|$ de \mathbb{R}^d .

Vitesse (ordre) de convergence

- **Convergence linéaire (ordre 1):** Si $K_1 = \lim_{n \rightarrow +\infty} \frac{\|x_{n+1} - x^*\|}{\|x_n - x^*\|}$ existe et $K_1 \in [0, 1[$, on dit que la suite converge **linéairement** vers x^* , ou que la convergence est d'ordre 1.
- **Convergence quadratique (ordre 2):** Si $K_1 = 0$, $K_2 = \lim_{n \rightarrow +\infty} \frac{\|x_{n+1} - x^*\|}{\|x_n - x^*\|^2}$ existe et non nul, on dit que la suite converge **quadratiquement** vers x^* , ou que la convergence est d'ordre 2.

- **Convergence d'ordre q** : Si $K_q = \lim_{n \rightarrow +\infty} \frac{\|x_{n+1} - x^*\|}{\|x_n - x^*\|^q}$ existe et $\neq 0$, la convergence est d'ordre q .

Remark 3.1.2. La constante K_1 est appelée constante asymptotique d'erreur pour la convergence linéaire.

Exemples

Example 3.1.3. Soit $x_n = (0.2)^n$. On a $\lim_{n \rightarrow +\infty} x_n = 0$. La convergence est vers $x^* = 0$.

$$\lim_{n \rightarrow +\infty} \frac{\|x_{n+1} - x^*\|}{\|x_n - x^*\|} = \lim_{n \rightarrow +\infty} \frac{(0.2)^{n+1}}{(0.2)^n} = 0.2 \in [0, 1[$$

D'où, x_n converge à l'ordre 1. Sa constante asymptotique est $K_1 = 0.2$.

Example 3.1.4. Soit $y_n = (0.2)^{2^n}$.

$$y_{n+1} = (0.2)^{2^{n+1}} = (0.2)^{2^n \cdot 2} = ((0.2)^{2^n})^2 = (y_n)^2$$

$$\lim_{n \rightarrow +\infty} \frac{\|y_{n+1} - x^*\|}{\|y_n - x^*\|^2} = \lim_{n \rightarrow +\infty} \frac{y_{n+1}}{(y_n)^2} = \lim_{n \rightarrow +\infty} \frac{(y_n)^2}{(y_n)^2} = 1$$

D'où, convergence d'ordre 2, de constante $K_2 = 1$.

Définition formelle de la convergence d'ordre q

Definition 3.1.5. On dit que x_n converge vers x^* à l'ordre q s'il existe un entier $N \in \mathbb{N}$ et des constantes $A, B \in \mathbb{R}$ telles que pour tout $n > N$,

$$0 < A \leq \frac{\|x_{n+1} - x^*\|}{\|x_n - x^*\|^q} \leq B < +\infty$$

Remark 3.1.6. La convergence est au moins d'ordre q si seulement on a

$$\limsup_{n \rightarrow +\infty} \frac{\|x_{n+1} - x^*\|}{\|x_n - x^*\|^q} < +\infty$$

3.1.3 Interprétation pratique de la vitesse de convergence

Nombre de chiffres significatifs

Remark 3.1.7. Si $|x_n - x^*| = 4 \cdot 10^{-8} = 0.\underbrace{00000004}_{8 \text{ digits}}$, on dit que x_n et x^* ont 8 chiffres exacts après la virgule.

$$\log_{10} |x_n - x^*| = \log_{10}(4 \cdot 10^{-8}) = \log_{10} 4 - 8 \approx -8$$

On pose $d_n = -\log_{10} |x_n - x^*|$. d_n mesure approximativement le nombre de chiffres décimaux exacts entre x_n et x^* .

$$\lim_{n \rightarrow +\infty} \frac{\|x_{n+1} - x^*\|}{\|x_n - x^*\|^q} = K_q \Rightarrow \|x_{n+1} - x^*\| \approx K_q \|x_n - x^*\|^q$$

$$\begin{aligned} \log_{10} \|x_{n+1} - x^*\| &\approx \log_{10}(K_q \|x_n - x^*\|^q) \\ &= \log_{10} K_q + q \log_{10} \|x_n - x^*\| \end{aligned}$$

$$\begin{aligned} -d_{n+1} &\approx \log_{10} K_q + q(-d_n) \\ d_{n+1} &\approx qd_n - \log_{10} K_q \end{aligned}$$

Si $q = 1$, $d_{n+1} \approx d_n - \log_{10} K_1$. À chaque itération, on gagne environ $-\log_{10} K_1$ chiffres significatifs.

Si $q > 1$, $d_{n+1} \approx qd_n$. Le nombre de chiffres significatifs est approximativement multiplié par q à chaque itération.

3.1.4 Nombre d'itérations pour gagner un chiffre en convergence linéaire

Proposition 3.1.8. Si x_n converge à l'ordre 1 vers x^* avec une constante asymptotique K_1 . Alors, le nombre d'itérations nécessaires pour gagner un chiffre significatif est approximativement $-\frac{1}{\log_{10}(K_1)}$.

Preuve. Soit m le nombre d'itérations pour gagner 1 chiffre significatif. Pour une convergence linéaire, $d_{n+1} \approx d_n - \log_{10} K_1$. En partant de d_n , après m itérations, on aura:

$$d_{n+m} \approx d_n - m \log_{10} K_1$$

On souhaite gagner 1 chiffre significatif, donc $d_{n+m} \approx d_n + 1$.

$$d_n + 1 = d_n - m \log_{10} K_1 \Leftrightarrow 1 = -m \log_{10} K_1 \Leftrightarrow m = -\frac{1}{\log_{10} K_1}$$

□

3.1.5 Linéarisation pour estimation graphique de q

$$\begin{aligned} \frac{\|x_{n+1} - x^*\|}{\|x_n - x^*\|^q} &\approx K_q \\ \log_{10} \|x_{n+1} - x^*\| &\approx \log_{10}(K_q \|x_n - x^*\|^q) \\ \log_{10} \|x_{n+1} - x^*\| &\approx \underbrace{q \log_{10} \|x_n - x^*\|}_x + \underbrace{\log_{10} K_q}_b \end{aligned}$$

C'est de la forme $y = qx + b$, où $y = \log_{10} \|x_{n+1} - x^*\|$, $x = \log_{10} \|x_n - x^*\|$, q est la pente et $b = \log_{10} K_q$ est l'ordonnée à l'origine.

3.1.6 Procédure pour déterminer q graphiquement

1. Calculer les erreurs $\|x_n - x^*\|$ et $\|x_{n+1} - x^*\|$ pour les itérations disponibles.
2. Calculer les logarithmes (base 10) de ces erreurs: $\log_{10} \|x_n - x^*\|$ et $\log_{10} \|x_{n+1} - x^*\|$.
3. Tracer le nuage de points $(\log_{10} \|x_n - x^*\|, \log_{10} \|x_{n+1} - x^*\|)$.
4. Estimer graphiquement ou par régression linéaire la pente q de la droite qui approxime au mieux ce nuage de points. Cette pente q est une estimation de l'ordre de convergence.

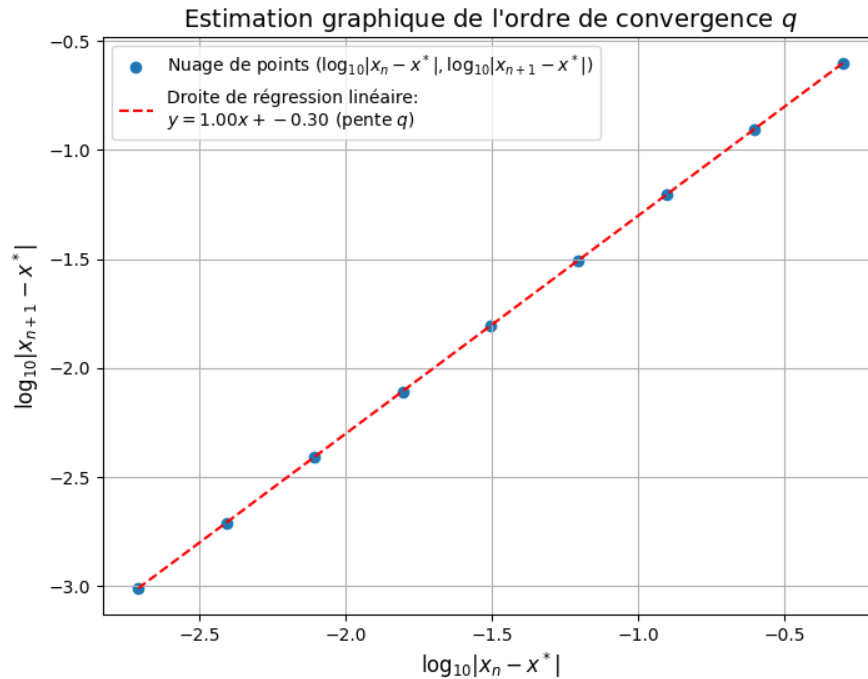


Figure 3.1: Estimation graphique de l'ordre de convergence q

3.2 Python code pour l'estimation de la convergence (exemple)

Listing 3.1: Code Python pour l'estimation de la convergence

```
import matplotlib.pyplot as plt
import matplotlib.patches as patches
import numpy as np

# Exemple de suite xn (remplacez avec votre suite)
xn = np.array([0.7**(i) for i in range(1, 20)])
x_star = 0 # Valeur limite de la suite

errors_n = np.abs(xn - x_star) # Erreur absolue |xn - x*|
log_errors_n = np.log10(errors_n) # Logarithme base 10 des erreurs

ex = log_errors_n[:-1] #  $\log_{10} |x_n - x^*|$ 
ey = log_errors_n[1:] #  $\log_{10} |x_{n+1} - x^*|$ 

plt.figure(figsize=(8, 6))
plt.scatter(ex, ey, label="Nuage de points")

ab = np.polyfit(ex, ey, 1) # Regression lineaire (y = ax + b)
y_fit = ab[0]*ex + ab[1]
plt.plot(ex, y_fit, color='red', linestyle='--', label=f"Droite de régression:  $y = {ab[0]:.2f}x + {ab[1]:.2f}$ ")
```



```

plt.xlabel("$\log_{10} \backslash |x_n - x^*|$", fontsize=12)
plt.ylabel("$\log_{10} \backslash |x_{n+1} - x^*|$", fontsize=12)
plt.title("Estimation graphique de l'ordre de convergence $q$", fontsize
    ↪ =14)
plt.legend()
plt.grid(True)
plt.show() # Affiche le graphique (pour execution hors LaTeX)

```

Chapter 4

CM4

4.1 Introduction à l'interpolation

4.1.1 Définition

Definition 4.1.1. Soit un nuage de points (exemple: un ensemble discret de points du graphe d'une fonction). Interpréter ce nuage de points correspond à chercher un polynôme de degré $N - 1$ qui passe par chacun de ces points.

- Comment le construire ?
- $P_{N-1} \in \mathbb{P}_{N-1}[x]$
- $P_{N-1}(x_i) = y_i$

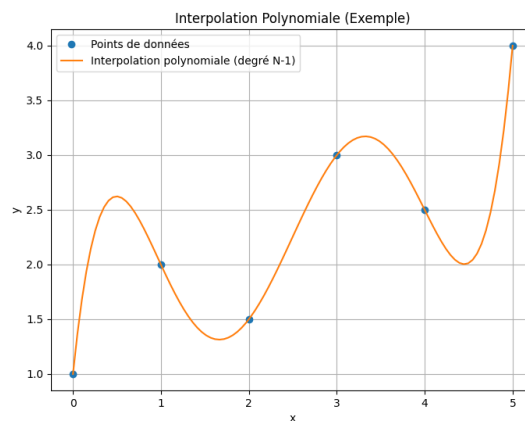


Figure 4.1: Illustration de l'interpolation polynomiale.

4.1.2 Motivations

- La solution d'un problème est fournie par une formule représentative : noyau de la chaleur (ex: convolution) et on cherche la solution en un nombre de points. \implies on approche alors la fonction par un polynôme i.e. chercher le polynôme de degré "bas" proche de la fonction.
- La solution d'un problème n'est connue qu'à travers ses valeurs en un nombre fini de points et on souhaite l'évaluer partout. \implies Interpolation.

- On peut utiliser l'interpolation dans :
 - la résolution numérique
 - la résolution numérique des Équations Différentielles Ordinaires (EDO)
 - la visualisation scientifique

Definition 4.1.2. Un tel polynôme est appelé **polynôme interpolateur de Lagrange** de degré $N - 1$ de ces points.

4.1.3 Exemples d'interpolation

Théorème: Polynôme interpolateur de degré 1

Theorem 4.1.3. Soient (x_1, y_1) et (x_2, y_2) deux points distincts de \mathbb{R}^2 . Il existe une unique droite D passant par ces deux points.

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

Si de plus, $x_1 \neq x_2$, il existe un unique polynôme de degré 1 (i.e., $P \in \mathbb{P}_1[x]$) tel que $y = P(x)$.
avec

$$P_1(x) = \frac{(x - x_2)}{(x_1 - x_2)} y_1 + \frac{(x - x_1)}{(x_2 - x_1)} y_2$$

pour des abscisses x_1, x_2 distinctes.

Example 4.1.4. Montrons que $M(x, y)$ est sur la droite $(M_1 M_2)$ si et seulement si les vecteurs $\overrightarrow{M_1 M}$ et $\overrightarrow{M_1 M_2}$ sont colinéaires. Soient $M_1(x_1, y_1)$, $M_2(x_2, y_2)$ et $M(x, y)$.

$$M \in (M_1 M_2) \iff \overrightarrow{M_1 M} // \overrightarrow{M_1 M_2}$$

$$\iff \det(\overrightarrow{M_1 M}, \overrightarrow{M_1 M_2}) = 0$$

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

$$\iff (x - x_1)(y_2 - y_1) - (x_2 - x_1)(y - y_1) = 0$$

Si $x_1 \neq x_2$, alors on peut réécrire l'équation de la droite sous la forme $y = ax + b$:

$$\implies y - y_1 = \frac{(y_2 - y_1)}{(x_2 - x_1)} (x - x_1)$$

$$\iff y = P_1(x)$$

Remark 4.1.5. On a l'écriture équivalente de P_1 :

$$P_1(x) = \frac{x - x_2}{x_1 - x_2} y_1 + \frac{x - x_1}{x_2 - x_1} y_2 = \frac{y_1 - y_2}{x_1 - x_2} x + \frac{x_2 y_1 - x_1 y_2}{x_2 - x_1} = a_1 x + a_0$$

c'est l'écriture dans la base $(1, x)$ de $\mathbb{P}_1[x]$ (base canonique).

$$\begin{aligned} \bullet P_1(x) &= \frac{x - x_2}{x_1 - x_2} y_1 + \frac{x - x_1}{x_2 - x_1} y_2 \\ &= \underbrace{\frac{x - x_2}{x_1 - x_2}}_{\ell_1(x)} y_1 + \underbrace{\frac{x - x_1}{x_2 - x_1}}_{\ell_2(x)} y_2 \end{aligned}$$

C'est l'écriture dans la base (ℓ_1, ℓ_2) de $\mathbb{P}_1[x]$ (base de Lagrange).

Remark 4.1.6. $\ell_1(x_1) = 1, \ell_1(x_2) = 0$
 $\ell_2(x_1) = 0, \ell_2(x_2) = 1$

- $P_1(x) = y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$ C'est l'écriture dans la base $(1, x - x_1)$ de $\mathbb{P}_1[x]$ (base de Newton).

Exemple: méthode de calcul employée

Chercher le polynôme interpolateur de Lagrange aux points $(x_1, y_1), (x_2, y_2), (x_3, y_3)$.

Méthode 1. $(x_1 \neq x_2, x_2 \neq x_3, x_1 \neq x_3)$

P_2 sera un polynôme de degré 2 :

$$P_2(x) = a_0 + a_1x + a_2x^2$$

Comme $P_2(x_i) = y_i$, pour $i = 1, 2, 3$, on a le système d'équations linéaires:

$$\begin{cases} P_2(x_1) = y_1 \\ P_2(x_2) = y_2 \\ P_2(x_3) = y_3 \end{cases} \implies \begin{cases} a_0 + a_1x_1 + a_2x_1^2 = y_1 \\ a_0 + a_1x_2 + a_2x_2^2 = y_2 \\ a_0 + a_1x_3 + a_2x_3^2 = y_3 \end{cases}$$

Matriciellement :

$$\begin{pmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \implies \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{pmatrix}^{-1}}_{H^{-1}} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

Matrice de Vandermonde mal-conditionnée mais facile à construire.

Remark 4.1.7. Pour 2 points :

$$H = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \end{pmatrix} \implies H^{-1} = \frac{1}{x_2 - x_1} \begin{pmatrix} x_2 & -x_1 \\ -1 & 1 \end{pmatrix}$$

si $x_1 \neq x_2$.

Méthode 2. Base de Newton

$$P_2(x) = a_0 + a_1(x - x_1) + a_2(x - x_1)(x - x_2)$$

$$\begin{cases} P_2(x_1) = y_1 \implies a_0 = y_1 \\ P_2(x_2) = y_2 \implies a_0 + a_1(x_2 - x_1) = y_2 \\ P_2(x_3) = y_3 \implies a_0 + a_1(x_3 - x_1) + a_2(x_3 - x_1)(x_3 - x_2) = y_3 \end{cases}$$

$$\implies \begin{cases} a_0 = y_1 \\ a_1 = \frac{y_2 - y_1}{x_2 - x_1} \\ a_2 = \frac{y_3 - y_1 - \frac{y_2 - y_1}{x_2 - x_1}(x_3 - x_1)}{(x_3 - x_1)(x_3 - x_2)} = \frac{\frac{y_3 - y_1}{x_3 - x_1} - \frac{y_2 - y_1}{x_2 - x_1}}{(x_3 - x_2)} = \frac{\frac{y_3 - y_1}{x_3 - x_1} - \frac{y_2 - y_1}{x_2 - x_1}}{x_3 - x_2} \end{cases}$$

Cette construction est différentielle et facile à mettre à jour quand on rajoute un point supplémentaire (on rajoute uniquement une ligne).

On a donc :

$$a_0 = y_1, \quad a_1 = \frac{y_2 - y_1}{x_2 - x_1}, \quad a_2 = \frac{\frac{y_3 - y_1}{x_3 - x_1} - \frac{y_2 - y_1}{x_2 - x_1}}{x_3 - x_2}$$

Le polynôme P_2 s'écrit donc :

$$P_2(x) = y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_1) + \frac{\frac{y_3 - y_1}{x_3 - x_1} - \frac{y_2 - y_1}{x_2 - x_1}}{x_3 - x_2}(x - x_1)(x - x_2)$$

	a_0	a_1	a_2
x_1	y_1		
x_2	y_1	$\frac{y_2 - y_1}{x_2 - x_1}$	
x_3	y_1	$\frac{y_2 - y_1}{x_2 - x_1}$	$\frac{\frac{y_3 - y_1}{x_3 - x_1} - \frac{y_2 - y_1}{x_2 - x_1}}{x_3 - x_2}$

Table 4.1: Tableau des coefficients pour la base de Newton.

Construction facile et différentielle par différences divisées : ajout d'un terme.

Méthode 3. Base de Lagrange

$$P_2(x) = \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)}y_1 + \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)}y_2 + \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)}y_3$$

$$P_2(x) = \sum_{i=1}^3 \left(\prod_{\substack{j=1 \\ j \neq i}}^3 \frac{x - x_j}{x_i - x_j} \right) y_i$$

$$= \ell_1(x)y_1 + \ell_2(x)y_2 + \ell_3(x)y_3$$

Remark 4.1.8. Pour deux points (x_1, y_1) et (x_2, y_2) , le polynôme interpolateur de Lagrange de degré 1 est :

$$P_1(x) = \frac{x - x_2}{x_1 - x_2}y_1 + \frac{x - x_1}{x_2 - x_1}y_2$$

4.2 Polynôme interpolateur de Lagrange

4.2.1 Définitions et propriétés

Théorème: Existence et unicité

Theorem 4.2.1 (Existence et unicité). Soient x_1, \dots, x_n des réels deux à deux distincts et y_1, \dots, y_n des réels quelconques. Il existe un unique polynôme $P \in \mathbb{P}_{n-1}[x]$ (i.e. de degré au plus $n - 1$) tel que $P(x_i) = y_i, \forall i = 1, \dots, n$.

On dit que P est le **polynôme interpolateur de Lagrange** aux points $(x_1, y_1), \dots, (x_n, y_n)$.

Preuve: Soit l'application linéaire $\Phi : \mathbb{P}_{n-1}[x] \rightarrow \mathbb{R}^n$ définie par

$$P \mapsto \begin{pmatrix} P(x_1) \\ \vdots \\ P(x_n) \end{pmatrix}$$

Montrons que Φ est injective. Si $\Phi(P) = 0$, alors $P(x_i) = 0$ pour tout $i = 1, \dots, n$. Donc P a n racines distinctes x_1, \dots, x_n . Comme P est un polynôme de degré au plus $n - 1$ avec n racines, il s'ensuit que $P \equiv 0$. Donc Φ est injective.

Comme $\mathbb{P}_{n-1}[x]$ et \mathbb{R}^n sont deux espaces vectoriels de même dimension n , une application linéaire injective est aussi bijective, donc un isomorphisme d'espaces vectoriels. La bijectivité de Φ assure l'existence et l'unicité du polynôme interpolateur.

Définition 4.2.2. Si f est une fonction continue sur $[a, b] \rightarrow \mathbb{R}$, et $x_1, \dots, x_n \in [a, b]$ sont n points deux à deux distincts, alors l'unique polynôme $P \in \mathbb{P}_{n-1}[x]$ tel que $P(x_i) = f(x_i)$, pour $i = 1, \dots, n$ est appelé **polynôme d'interpolation de Lagrange** de f aux points x_1, \dots, x_n .

4.2.2 Estimation de l'erreur d'interpolation

Théorème: Erreur d'interpolation

Theorem 4.2.3 (Erreur d'interpolation). Soient $a < b$, $f : [a, b] \rightarrow \mathbb{R}$ une fonction continue, et x_1, \dots, x_n n points deux à deux distincts dans $[a, b]$. Soit P_n le polynôme d'interpolation de Lagrange de f aux points x_i . Si f est de classe \mathcal{C}^n sur $[a, b]$, alors pour tout $x \in [a, b]$, il existe $\xi \in [a, b]$ tel que :

$$f(x) - P_n(x) = \frac{f^{(n)}(\xi)}{n!} \underbrace{\omega_n(x)}_{=\prod_{i=1}^n (x-x_i)}$$

où $\omega_n(x) = (x - x_1) \cdots (x - x_n)$.

Corollaire: Si $|f^{(n)}(x)|$ est bornée par M sur $[a, b]$ pour tout $x \in [a, b]$, alors $\forall x \in [a, b]$,

$$|f(x) - P_n(x)| \leq \frac{M}{n!} |\omega_n(x)| \leq \frac{M}{n!} (b - a)^n$$

Preuve: (à faire)

4.2.3 Implémentation avec Python

```
from scipy.interpolate import lagrange

x = np.array([1, 2, 3]) #à remplacer par les valeurs de x_1, x_2, x_3
y = np.array([2, 3, 1]) #à remplacer par les valeurs de y_1, y_2, y_3
p = lagrange(x,y)
print(p) # affiche le polynôme
print(p(2.5)) # évalue le polynôme en x=2.5
```

4.3 Construction des polynômes d'interpolation de Lagrange

4.3.1 Interpolation dans la base canonique (Vandermonde)

Construction

Soit $P(x) = \sum_{i=0}^{n-1} a_i x^i \in \mathbb{P}_{n-1}[x]$. On cherche les coefficients a_0, \dots, a_{n-1} tels que $P(x_k) = y_k$ pour $k = 1, \dots, n$.

$$\sum_{i=0}^{n-1} a_i x_k^i = y_k, \quad k = 1, \dots, n$$

Ce qui conduit au système linéaire matriciel suivant :

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

$$V(x_1, \dots, x_n) \mathbf{a} = \mathbf{y}$$

où $V(x_1, \dots, x_n)$ est la matrice de Vandermonde.

C'est une matrice pleine, souvent mal conditionnée, mais facile à construire.

```

def VDM_Mat(x):
    n = len(x)
    V = np.zeros((n, n))
    for i in range(n):
        for j in range(n):
            V[i, j] = x[i]**j
    return V

def VDM_Poly(x,y):
    M = VDM_Mat(x)
    a = np.linalg.solve(M, y)
    return a

```

4.3.2 Evaluation efficace : Algorithme de Horner

Proposition: Algorithme de Horner

Proposition 4.3.1. Soit $P(x) = a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n$ un polynôme. On définit la suite $(q_k)_{k=0}^n$ par :

$$\begin{cases} q_0 = a_0 \\ q_k = q_{k-1}x + a_k, \quad k = 1, \dots, n \end{cases}$$

Alors $q_n = P(x)$.

Exemple: $P(x) = x^2 + 2x + 1 = (x + 1)^2$

Pour évaluer $P(2)$: $q_0 = a_0 = 1$ $q_1 = q_0 \times 2 + a_1 = 1 \times 2 + 2 = 4$ $q_2 = q_1 \times 2 + a_2 = 4 \times 2 + 1 = 9 = P(2) = 2^2 + 2 \times 2 + 1 = 9$

```

def Horner(P, xx):
    y = 0
    for a in P:
        y = y*xx + a
    return y

def IntVal_VDM (x, y, xx):
    a = VDM_Poly(x,y)
    YY = Horner(a[::-1], xx) # reverse a pour correspondre à l'ordre des
    ↪ coefficients dans Horner
    return YY

```

4.3.3 Interpolation dans la base duale: Formule de Lagrange et points barycentriques

Construction

L'idée est de prendre pour base de $\mathbb{P}_{n-1}[x]$ l'image réciproque de la base canonique de \mathbb{R}^n par l'application Φ définie dans le théorème d'existence et unicité. On cherche donc une base $\{\mathcal{L}_j\}_{j=1}^n$ de $\mathbb{P}_{n-1}[x]$ telle que

$$\mathcal{L}_j(x_i) = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases}$$

On construit les polynômes de Lagrange $\mathcal{L}_j(x)$ comme suit :

$$\mathcal{L}_j(x) = \prod_{\substack{i=1 \\ i \neq j}}^n \frac{x - x_i}{x_j - x_i} = \frac{\prod_{i \neq j} (x - x_i)}{\prod_{i \neq j} (x_j - x_i)}$$

Le polynôme interpolateur de Lagrange s'écrit alors :

$$P(x) = \sum_{j=1}^n y_j \mathcal{L}_j(x)$$

Chapter 5

CM5

5.1 Introduction à l'interpolation polynomiale

L'interpolation polynomiale est une technique fondamentale en analyse numérique qui consiste à trouver un polynôme qui passe par un ensemble donné de points. Plus précisément, étant donnés $n + 1$ points $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ où les x_i sont distincts, l'interpolation polynomiale cherche à construire un polynôme $P(x)$ de degré au plus n tel que $P(x_i) = y_i$ pour $i = 0, 1, \dots, n$. Ce polynôme $P(x)$ est appelé le polynôme d'interpolation.

L'interpolation polynomiale a de nombreuses applications dans divers domaines tels que l'approximation de fonctions, l'intégration numérique, la résolution d'équations différentielles et le traitement de données expérimentales. Différentes méthodes existent pour construire ce polynôme d'interpolation, chacune ayant ses avantages et ses inconvénients en termes de complexité, de stabilité et de facilité d'implémentation. Nous allons explorer ici les méthodes de Lagrange et de Newton.

5.2 Interpolation de Lagrange

5.2.1 Formule de Lagrange

La formule d'interpolation de Lagrange est une manière explicite d'écrire le polynôme d'interpolation. Elle repose sur l'utilisation des polynômes de base de Lagrange.

Definition 5.2.1 (Polynômes de base de Lagrange). Soient x_0, x_1, \dots, x_n des points distincts. Le polynôme nodal $\omega_{n+1}(x)$ est défini par :

$$\omega_{n+1}(x) = \prod_{i=0}^n (x - x_i)$$

et les polynômes de base de Lagrange $l_i(x)$ associés aux points x_0, x_1, \dots, x_n sont définis par :

$$l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j} = \frac{\omega_{n+1}(x)}{(x - x_i)\omega'_{n+1}(x_i)}$$

pour $i = 0, 1, \dots, n$.

On remarque que les polynômes de base de Lagrange vérifient la propriété suivante :

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

Proposition 5.2.2 (Formule d'interpolation de Lagrange). Le polynôme d'interpolation de Lagrange $P(x)$ de degré au plus n qui interpole les points $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ est donné par :

$$P(x) = \sum_{i=0}^n y_i l_i(x)$$

Preuve. Pour vérifier que $P(x)$ est bien le polynôme d'interpolation, il suffit de montrer que $P(x_j) = y_j$ pour tout $j = 0, 1, \dots, n$.

$$P(x_j) = \sum_{i=0}^n y_i l_i(x_j) = \sum_{i=0}^n y_i \delta_{ij} = y_j$$

De plus, chaque $l_i(x)$ est un polynôme de degré n , donc $P(x)$ est un polynôme de degré au plus n . \square

5.2.2 Exemple

[Insérer un exemple si disponible dans les notes manuscrites, sinon, un exemple simple peut être construit ici.]

5.3 Erreur d'interpolation

5.3.1 Formule de l'erreur d'interpolation

L'erreur d'interpolation mesure la différence entre la fonction $f(x)$ que l'on cherche à interpoler et le polynôme d'interpolation $P(x)$.

Proposition 5.3.1 (Formule de l'erreur d'interpolation). Soit $f \in C^{n+1}([a, b])$ et $P(x)$ le polynôme d'interpolation de Lagrange de degré au plus n interpolant f aux points $x_0, x_1, \dots, x_n \in [a, b]$. Alors, pour tout $x \in [a, b]$, il existe un point $\xi_x \in [a, b]$ tel que :

$$f(x) - P(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \omega_{n+1}(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{i=0}^n (x - x_i)$$

5.3.2 Analyse de l'erreur

La formule de l'erreur montre que l'erreur d'interpolation dépend de deux facteurs principaux :

- La dérivée $(n+1)$ -ième de la fonction f , $f^{(n+1)}(\xi_x)$. Si la dérivée $(n+1)$ -ième de f est petite sur $[a, b]$, alors l'erreur d'interpolation sera petite.
- Le polynôme nodal $\omega_{n+1}(x) = \prod_{i=0}^n (x - x_i)$. La distribution des points d'interpolation x_0, x_1, \dots, x_n influence la magnitude de $\omega_{n+1}(x)$. Choisir des points d'interpolation de manière à minimiser $|\omega_{n+1}(x)|$ sur $[a, b]$ peut réduire l'erreur d'interpolation. Par exemple, les points de Chebyshev sont connus pour minimiser la norme infinie de $\omega_{n+1}(x)$ sur $[-1, 1]$.

5.3.3 Convergence

Pour assurer la convergence de l'interpolation polynomiale, c'est-à-dire que $P_n(x) \rightarrow f(x)$ lorsque $n \rightarrow \infty$, il ne suffit pas d'augmenter le degré du polynôme d'interpolation en utilisant des points équidistants. Le phénomène de Runge montre que pour certaines fonctions, l'interpolation polynomiale avec des points équidistants peut diverger entre les nœuds, même si la fonction est analytique. Cependant, si on choisit judicieusement les points d'interpolation, comme les points de Chebyshev, et si la fonction f est suffisamment régulière, on peut garantir la convergence de l'interpolation polynomiale.

5.4 Interpolation de Newton

5.4.1 Différences divisées

La méthode de Newton utilise les différences divisées pour construire le polynôme d'interpolation.

Definition 5.4.1 (Différences divisées). Les différences divisées d'ordre zéro sont définies par $f[x_i] = f(x_i)$. Les différences divisées d'ordre supérieur sont définies par la formule de récurrence :

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

Proposition 5.4.2 (Formule d'interpolation de Newton). Le polynôme d'interpolation de Newton de degré au plus n s'écrit :

$$P(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \dots + f[x_0, x_1, \dots, x_n] \prod_{i=0}^{n-1} (x - x_i)$$

que l'on peut écrire sous la forme :

$$P(x) = \sum_{k=0}^n f[x_0, x_1, \dots, x_k] \prod_{i=0}^{k-1} (x - x_i)$$

avec $\prod_{i=0}^{-1} (x - x_i) = 1$.

5.4.2 Algorithme de calcul des différences divisées

Les différences divisées peuvent être organisées dans un tableau triangulaire.

Listing 5.1: Calcul des différences divisées

```
import matplotlib.pyplot as plt
import matplotlib.patches as patches
import numpy as np

def DifferencesDivisees(x,y):
    n = len(x)
    d = np.zeros([n, n])
    for i in range(n):
        d[i, 0] = y[i]
    for j in range(1, n):
        for i in range(n - j):
            d[i, j] = (d[i+1, j-1] - d[i, j-1]) / (x[i+j] - x[i])
    return d
```

Remark 5.4.3. La fonction `DifferencesDivisees(x,y)` prend en entrée les abscisses x et les ordonnées y des points d'interpolation et retourne une matrice d contenant les différences divisées. La diagonale supérieure de cette matrice contient les coefficients $f[x_0], f[x_0, x_1], \dots, f[x_0, x_1, \dots, x_n]$ nécessaires pour la formule d'interpolation de Newton.

5.4.3 Évaluation du polynôme de Newton : Formule de Horner-Newton

Pour évaluer efficacement le polynôme de Newton, on utilise la formule de Horner-Newton.

Listing 5.2: Évaluation du polynôme de Newton (Horner-Newton)

```
import matplotlib.pyplot as plt
import matplotlib.patches as patches
import numpy as np

def HornerNewton(a,x,xx):
    n = len(a)
    yy = a[n-1]
    for i in range(n-2, -1, -1):
        yy = a[i] + (xx - x[i]) * yy
    return yy
```

Remark 5.4.4. La fonction `HornerNewton(a,x,xx)` prend en entrée le vecteur des coefficients des différences divisées `a` (diagonale de la matrice retournée par `DifferencesDivisees`), le vecteur des abscisses `x` des points d'interpolation, et un point `xx` où l'on souhaite évaluer le polynôme. Elle retourne la valeur du polynôme de Newton évalué en `xx`. Cette méthode est plus efficace pour évaluer le polynôme que l'évaluation directe de la formule de Newton.

5.5 Comparaison des méthodes et complexité

5.5.1 Complexité

- **Interpolation de Lagrange:** Le calcul des polynômes de base de Lagrange $l_i(x)$ et l'évaluation du polynôme d'interpolation de Lagrange nécessitent $\mathcal{O}(n^2)$ opérations pour un point donné x . Si l'on souhaite obtenir la forme développée du polynôme, la complexité est plus élevée.
- **Interpolation de Newton:** Le calcul des différences divisées nécessite $\mathcal{O}(n^2)$ opérations. L'évaluation du polynôme de Newton en utilisant la formule de Horner-Newton nécessite $\mathcal{O}(n)$ opérations par point. C'est une méthode efficace pour évaluer le polynôme une fois les différences divisées calculées.

5.5.2 Optimisation et ajout de points

- **Formule de Horner pour Newton:** La formule de Horner-Newton est cruciale pour l'efficacité de l'évaluation du polynôme de Newton. Elle réduit la complexité de l'évaluation à $\mathcal{O}(n)$ une fois les coefficients (différences divisées) sont connus.
- **Ajout d'un nouveau point:**
 - *Lagrange:* L'ajout d'un nouveau point d'interpolation nécessite de recalculer tous les polynômes de base de Lagrange et de refaire la somme. Cela peut être coûteux.
 - *Newton:* Si on ajoute un nouveau point (x_{n+1}, y_{n+1}) , on peut facilement étendre le polynôme d'interpolation de Newton en calculant une différence divisée supplémentaire $f[x_0, x_1, \dots, x_{n+1}]$ et en ajoutant un terme à la formule existante. Les différences divisées déjà calculées restent valides. C'est un avantage majeur de la méthode de Newton.

5.6 Conclusion

L'interpolation polynomiale est un outil puissant pour approximer des fonctions et traiter des données. Les méthodes de Lagrange et Newton offrent différentes approches pour construire et évaluer le polynôme d'interpolation. La méthode de Lagrange donne une formule explicite mais est moins pratique pour les calculs et l'ajout de nouveaux points. La méthode de Newton, basée sur les différences divisées, est efficace pour l'évaluation grâce à la formule de Horner-Newton et permet d'ajouter facilement de nouveaux points d'interpolation. Le choix de la méthode dépend du contexte et des besoins spécifiques de l'application.

Chapter 6

CM6

6.1 Polynômes de Tchebychev

6.1.1 Définition

On définit les polynômes de Tchebychev par récurrence :

- $T_0(x) = 1$
- $T_1(x) = x$
- $T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad n \geq 1$

Les premiers polynômes de Tchebychev sont donc :

- $T_0(x) = 1$
- $T_1(x) = x$
- $T_2(x) = 2xT_1(x) - T_0(x) = 2x^2 - 1$
- $T_3(x) = 2xT_2(x) - T_1(x) = 2x(2x^2 - 1) - x = 4x^3 - 3x$

6.1.2 Expression trigonométrique

On a aussi l'expression trigonométrique suivante pour les polynômes de Tchebychev :

$$T_n(x) = \cos(n \arccos(x)), \quad x \in [-1, 1]$$

On vérifie pour $n = 0, 1, 2$:

- $T_0(x) = \cos(0 \arccos(x)) = \cos(0) = 1$
- $T_1(x) = \cos(1 \arccos(x)) = \cos(\arccos(x)) = x$
- $T_2(x) = \cos(2 \arccos(x)) = 2 \cos^2(\arccos(x)) - 1 = 2x^2 - 1$

Pour vérifier la relation de récurrence, posons $\theta = \arccos(x)$, donc $x = \cos(\theta)$. Alors

$$\begin{aligned} 2xT_n(x) - T_{n-1}(x) &= 2 \cos(\theta) \cos(n\theta) - \cos((n-1)\theta) \\ &= \cos((n+1)\theta) + \cos((n-1)\theta) - \cos((n-1)\theta) \\ &= \cos((n+1)\theta) \\ &= T_{n+1}(x) \end{aligned}$$

On a utilisé la formule trigonométrique : $\cos(a) \cos(b) = \frac{1}{2}[\cos(a+b) + \cos(a-b)]$.

6.1.3 Propriétés

1. **Racines de $T_n(x)$:** $T_n(x) = 0 \Leftrightarrow \cos(n \arccos(x)) = 0$. Posons $x = \cos(\theta)$. Alors $\cos(n\theta) = 0 \Leftrightarrow n\theta = \frac{\pi}{2} + k\pi, k \in \mathbb{Z}$. Donc $\theta = \frac{\pi}{2n} + k\frac{\pi}{n}$. Pour avoir n racines distinctes dans $[-1, 1]$, on prend $k = 0, 1, \dots, n-1$.

$$x_k = \cos\left(\frac{\pi}{2n} + k\frac{\pi}{n}\right), \quad k = 0, 1, \dots, n-1$$

sont les n racines de $T_n(x)$ dans $[-1, 1]$.

2. $|T_n(x)| \leq 1$ pour $x \in [-1, 1]$. En effet, pour $x \in [-1, 1]$, $T_n(x) = \cos(n \arccos(x))$, et $|\cos(\cdot)| \leq 1$. De plus, $T_n(\cos(\frac{k\pi}{n})) = \cos(k\pi) = (-1)^k$. Donc $\max_{x \in [-1, 1]} |T_n(x)| = 1$ atteint en $x'_k = \cos(\frac{k\pi}{n})$, $k = 0, \dots, n$.

3. **Orthogonalité:** Les polynômes de Tchebychev sont orthogonaux pour le produit scalaire :

$$\langle f, g \rangle = \int_{-1}^1 \frac{f(x)g(x)}{\sqrt{1-x^2}} dx$$

En effet, posons $x = \cos(\theta)$, $dx = -\sin(\theta)d\theta$, $\sqrt{1-x^2} = \sin(\theta)$.

$$\int_{-1}^1 \frac{T_n(x)T_m(x)}{\sqrt{1-x^2}} dx = \int_{\pi}^0 \frac{\cos(n\theta)\cos(m\theta)}{\sin(\theta)} (-\sin(\theta)) d\theta = \int_0^{\pi} \cos(n\theta)\cos(m\theta) d\theta$$

On sait que $\int_0^{\pi} \cos(n\theta)\cos(m\theta) d\theta = 0$ si $n \neq m$, et $\int_0^{\pi} \cos^2(n\theta) d\theta = \frac{\pi}{2}$ si $n \neq 0$, et $\int_0^{\pi} \cos^2(0) d\theta = \pi$.
Donc

$$\int_{-1}^1 \frac{T_n(x)T_m(x)}{\sqrt{1-x^2}} dx = \begin{cases} 0 & \text{si } n \neq m \\ \pi & \text{si } n = m = 0 \\ \frac{\pi}{2} & \text{si } n = m \neq 0 \end{cases}$$

6.1.4 Application : Polynôme de meilleure approximation uniforme

Proposition 6.1.1. Soit P un polynôme de degré n avec coefficient dominant égal à 1, alors

$$\max_{x \in [-1, 1]} |P(x)| \geq \max_{x \in [-1, 1]} \left| \frac{1}{2^{n-1}} T_n(x) \right| = \frac{1}{2^{n-1}}$$

De plus, il y a égalité si et seulement si $P(x) = \frac{1}{2^{n-1}} T_n(x)$. On dit que $\frac{1}{2^{n-1}} T_n(x)$ est le polynôme de Tchebychev normalisé.

Soient x_0, \dots, x_n des points 2 à 2 distincts de $[-1, 1]$. On a :

$$\max_{x \in [-1, 1]} \prod_{i=0}^n |x - x_i| \geq \max_{x \in [-1, 1]} \prod_{i=1}^n |x - x_i^*|$$

avec x_i^* les racines de $T_{n+1}(x)$ translatées et dilatées sur $[-1, 1]$ (racines de Tchebychev).

$$x_k^* = \cos\left(\frac{\pi}{2(n+1)} + \frac{k\pi}{n+1}\right), \quad k = 0, \dots, n$$

sont les racines de T_{n+1} .

6.1.5 Application à l'interpolation polynomiale

Soient x_0, \dots, x_n $n+1$ points 2 à 2 distincts, f une fonction $n+1$ fois continûment dérivable. Soit $P_n(x)$ le polynôme d'interpolation de Lagrange de f aux points x_i . Alors l'erreur d'interpolation est donnée par :

$$f(x) - P_n(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{i=0}^n (x - x_i), \quad \xi_x \in [\min(x, x_i), \max(x, x_i)]$$

Donc

$$|f(x) - P_n(x)| \leq \frac{\max_{\xi \in [a,b]} |f^{(n+1)}(\xi)|}{(n+1)!} \max_{x \in [a,b]} \prod_{i=0}^n |x - x_i|$$

Pour minimiser l'erreur d'interpolation, il faut minimiser $\max_{x \in [a,b]} \prod_{i=0}^n |x - x_i|$. D'après le corollaire précédent, les points de Tchebychev minimisent ce terme (à translation et dilatation près pour adapter l'intervalle $[-1, 1]$ à $[a, b]$).

6.2 Intégration numérique

6.2.1 Motivation et concept général

On cherche à approcher numériquement l'intégrale d'une fonction f sur un intervalle $[a, b]$:

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

On approche $I(f)$ par une somme pondérée de valeurs de f en certains points $x_i \in [a, b]$:

$$Q_n(f) = \sum_{i=0}^n \omega_i f(x_i)$$

où x_i sont les **nœuds** de quadrature et ω_i sont les **poinds** de quadrature. On cherche à construire des formules de quadrature $Q_n(f)$ qui soient exactes pour les polynômes de degré le plus élevé possible.

Définition 6.2.1. On dit qu'une formule de quadrature $Q_n(f) = \sum_{i=0}^n \omega_i f(x_i)$ est de degré de précision r si elle est exacte pour tous les polynômes de degré $\leq r$, et n'est pas exacte pour au moins un polynôme de degré $r + 1$. C'est-à-dire :

- $\forall P \in \mathbb{P}_r, \quad Q_n(P) = I(P) = \int_a^b P(x) dx$
- $\exists P \in \mathbb{P}_{r+1}, \quad Q_n(P) \neq I(P) = \int_a^b P(x) dx$

6.2.2 Construction des formules de quadrature

Idée : utiliser l'interpolation polynomiale. Soient x_0, \dots, x_n $n + 1$ points distincts dans $[a, b]$. Soit $P_n(x)$ le polynôme d'interpolation de Lagrange de f aux points x_i . On approche $I(f)$ par $I(P_n) = \int_a^b P_n(x) dx$. On sait que $P_n(x) = \sum_{i=0}^n f(x_i) L_i(x)$ où $L_i(x)$ sont les polynômes de Lagrange :

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

Donc

$$Q_n(f) = I(P_n) = \int_a^b P_n(x) dx = \int_a^b \sum_{i=0}^n f(x_i) L_i(x) dx = \sum_{i=0}^n f(x_i) \int_a^b L_i(x) dx$$

On pose $\omega_i = \int_a^b L_i(x) dx$. Alors $Q_n(f) = \sum_{i=0}^n \omega_i f(x_i)$ est une formule de quadrature.

Proposition 6.2.2. La formule de quadrature $Q_n(f)$ construite à partir de l'interpolation de Lagrange aux points x_0, \dots, x_n est de degré de précision au moins n .

Preuve. Si $P \in \mathbb{P}_n$, alors $P_n(x) = P(x)$ (le polynôme d'interpolation d'un polynôme de degré $\leq n$ est lui-même). Donc $Q_n(P) = \int_a^b P_n(x)dx = \int_a^b P(x)dx = I(P)$. Donc Q_n est exacte pour les polynômes de degré $\leq n$. \square

6.2.3 Exemples

Exemple 6.2.3 (Formule du point milieu). $n = 0$, un seul point $x_0 = \frac{a+b}{2}$ (milieu de l'intervalle). $L_0(x) = 1$. $\omega_0 = \int_a^b L_0(x)dx = \int_a^b 1dx = b - a$. $Q_0(f) = (b - a)f\left(\frac{a+b}{2}\right)$. Degré de précision : 1. Exacte pour les polynômes de degré ≤ 1 . Exemple : $\int_0^1 x dx = \frac{1}{2}$. $Q_0(x) = (1 - 0) \times \frac{0+1}{2} = \frac{1}{2}$. Exact. $\int_0^1 x^2 dx = \frac{1}{3}$. $Q_0(x^2) = (1 - 0) \times \left(\frac{0+1}{2}\right)^2 = \frac{1}{4} \neq \frac{1}{3}$. Non exacte pour degré 2.

Exemple 6.2.4 (Formule des trapèzes). $n = 1$, deux points $x_0 = a$, $x_1 = b$. $L_0(x) = \frac{x-x_1}{x_0-x_1} = \frac{x-b}{a-b}$, $L_1(x) = \frac{x-x_0}{x_1-x_0} = \frac{x-a}{b-a}$. $\omega_0 = \int_a^b \frac{x-b}{a-b} dx = \frac{1}{a-b} \left[\frac{x^2}{2} - bx \right]_a^b = \frac{1}{a-b} \left[\left(\frac{b^2}{2} - b^2 \right) - \left(\frac{a^2}{2} - ba \right) \right] = \frac{1}{a-b} \left[-\frac{b^2}{2} - \frac{a^2}{2} + ba \right] = \frac{b-a}{2}$. $\omega_1 = \int_a^b \frac{x-a}{b-a} dx = \frac{1}{b-a} \left[\frac{x^2}{2} - ax \right]_a^b = \frac{1}{b-a} \left[\left(\frac{b^2}{2} - ab \right) - \left(\frac{a^2}{2} - a^2 \right) \right] = \frac{1}{b-a} \left[\frac{b^2}{2} - ab + \frac{a^2}{2} \right] = \frac{b-a}{2}$. $Q_1(f) = \frac{b-a}{2} [f(a) + f(b)]$. Degré de précision : 1. Exacte pour les polynômes de degré ≤ 1 . Exemple : $\int_0^1 x^2 dx = \frac{1}{3}$. $Q_1(x^2) = \frac{1-0}{2} [0^2 + 1^2] = \frac{1}{2} \neq \frac{1}{3}$. Non exacte pour degré 2.

6.2.4 Estimation de l'erreur

Soit $Q_n(f) = \sum_{i=0}^n \omega_i f(x_i)$ la formule de quadrature construite par interpolation de Lagrange aux points x_0, \dots, x_n . On sait que l'erreur d'interpolation est :

$$f(x) - P_n(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{i=0}^n (x - x_i)$$

Donc l'erreur de quadrature est :

$$\begin{aligned} E_n(f) &= I(f) - Q_n(f) = \int_a^b f(x)dx - \int_a^b P_n(x)dx = \int_a^b [f(x) - P_n(x)]dx \\ &= \int_a^b \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{i=0}^n (x - x_i) dx = \frac{f^{(n+1)}(\xi)}{(n+1)!} \int_a^b \prod_{i=0}^n (x - x_i) dx \end{aligned}$$

Si on suppose que $f^{(n+1)}$ est continue, on peut utiliser la formule de la moyenne pour l'intégrale :

$$E_n(f) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \int_a^b \prod_{i=0}^n (x - x_i) dx, \quad \xi \in [a, b]$$

où ξ est une valeur intermédiaire dans $[a, b]$. En pratique, on borne l'erreur :

$$|E_n(f)| \leq \frac{\max_{\xi \in [a, b]} |f^{(n+1)}(\xi)|}{(n+1)!} \int_a^b \prod_{i=0}^n |x - x_i| dx$$

Remark 6.2.5. Pour la formule du point milieu sur $[-1, 1]$, $x_0 = 0$, $n = 0$. $Q_0(f) = 2f(0)$. $\int_{-1}^1 (x - x_0)dx = \int_{-1}^1 x dx = 0$. Donc la formule d'erreur simple ne s'applique pas directement car $\int_a^b \prod_{i=0}^n (x - x_i)dx = 0$ dans certains cas (comme ici). Il faut une formule d'erreur plus précise.



Chapter 7

CM7

7.1 Introduction

L'intégration numérique est essentielle pour approximer les intégrales définies, en particulier lorsque le calcul analytique est impossible ou complexe. Les formules de quadrature offrent une approche pour estimer ces intégrales en utilisant une somme pondérée de valeurs de la fonction en des points spécifiques. Ce chapitre introduit les formules de quadrature composites, qui consistent à appliquer des formules de quadrature élémentaires sur des subdivisions de l'intervalle d'intégration pour améliorer la précision de l'approximation. Nous explorerons les concepts de formules de quadrature élémentaires et composites, et nous détaillerons les méthodes classiques telles que les méthodes des rectangles, des trapèzes, du point milieu et de Simpson.

7.2 Formules de Quadrature Élémentaires

Definition 7.2.1 (Formule de Quadrature Élémentaire). On appelle formule de quadrature élémentaire sur l'intervalle $I_e = [-1, 1]$, associée aux points $x_i \in [-1, 1]$ et aux poids w_i , la formule :

$$I_e(f) = \sum_{i=1}^N w_i f(x_i)$$

qui approche l'intégrale $\int_{-1}^1 f(t)dt$ pour une fonction $f \in C^0([-1, 1])$.

Plus généralement, pour un intervalle $[a, b]$, on peut définir une formule de quadrature élémentaire.

Definition 7.2.2. Étant donné un intervalle $[a, b]$, une formule de quadrature élémentaire $I_e(f)$ est une approximation de l'intégrale $\int_a^b f(t)dt$ de la forme :

$$I_e(f) = \sum_{i=1}^N \omega_i f(\xi_i)$$

où $\xi_i \in [a, b]$ sont les points de quadrature et ω_i sont les poids associés.

7.3 Formules de Quadrature Composites

7.3.1 Principe de construction

Pour améliorer la précision de l'approximation, on utilise des formules de quadrature composites. L'idée est de subdiviser l'intervalle d'intégration $[a, b]$ en n sous-intervalles $[x_i, x_{i+1}]$ où $a = x_0 < x_1 < \dots < x_n = b$.

Sur chaque sous-intervalle $[x_i, x_{i+1}]$, on applique une formule de quadrature élémentaire. La formule de quadrature composite $I_c(f)$ sur $[a, b]$ est la somme des approximations sur chaque sous-intervalle :

$$I_c(f) = \sum_{i=0}^{n-1} I_e(f|_{[x_i, x_{i+1}]})$$

où $I_e(f|_{[x_i, x_{i+1}]})$ est la formule de quadrature élémentaire appliquée à la fonction f sur l'intervalle $[x_i, x_{i+1}]$.

Dans le cas d'une subdivision uniforme de l'intervalle $[a, b]$ en n sous-intervalles, on pose $h = \frac{b-a}{n}$ et $x_i = a + ih$ pour $i = 0, 1, \dots, n$.

7.3.2 Visualisation

La visualisation des formules de quadrature composites permet de comprendre géométriquement l'approximation de l'intégrale par des sommes d'aires de rectangles ou de trapèzes.

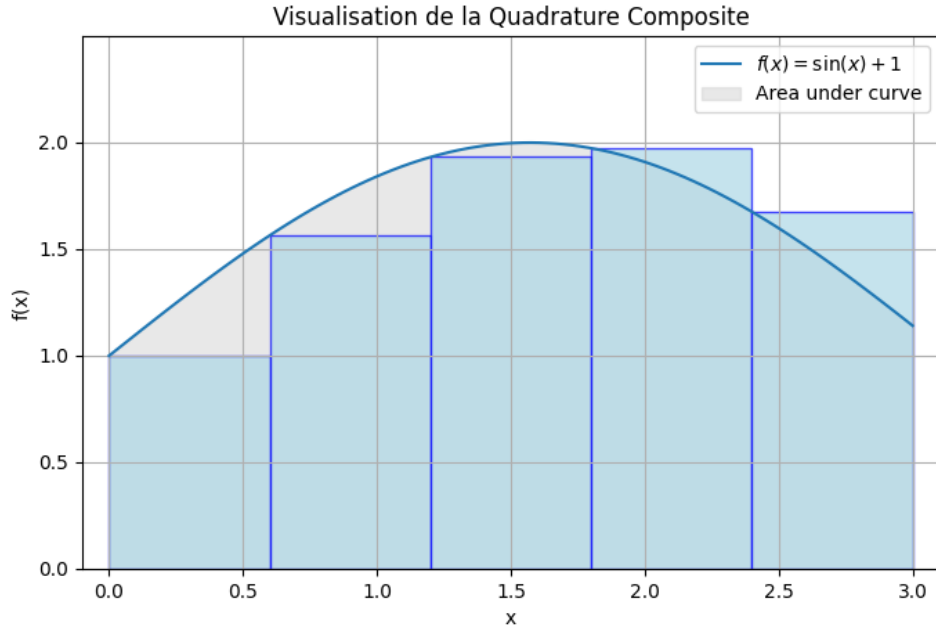


Figure 7.1: Visualisation d'une formule de quadrature composite (méthode des rectangles gauches).

Considérons une subdivision de l'intervalle $[a, b]$ en n intervalles $[x_j, x_{j+1}]$ de longueur $h = \frac{b-a}{n}$. La formule de quadrature composite s'écrit alors comme la somme des contributions sur chaque intervalle.

7.4 Méthodes de Quadrature Composites Classiques

7.4.1 Méthode des Rectangles

Définition

La méthode des rectangles approche l'intégrale sur chaque sous-intervalle $[x_j, x_{j+1}]$ par l'aire du rectangle de hauteur $f(x_j)$ (rectangle gauche) ou $f(x_{j+1})$ (rectangle droit), ou $f(\frac{x_j+x_{j+1}}{2})$ (point milieu). Pour la méthode des rectangles gauches élémentaire sur un intervalle $[\alpha, \beta]$, on a:

$$\int_{\alpha}^{\beta} f(t)dt \approx (\beta - \alpha)f(\alpha)$$

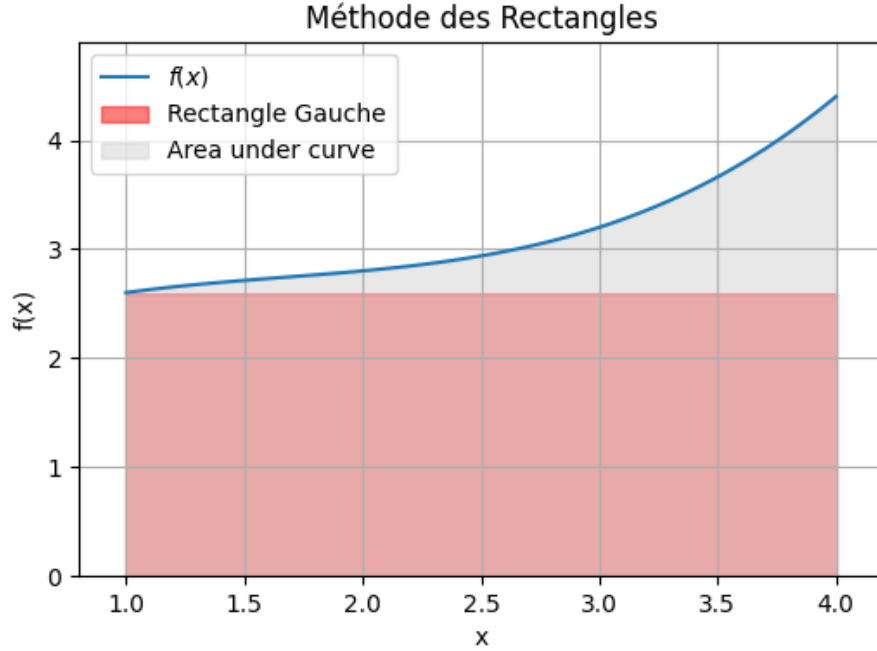


Figure 7.2: Méthode des Rectangles (gauche).

La formule de quadrature composite des rectangles gauches sur $[a, b]$ avec n intervalles est donnée par:

$$I_c(f) = h \sum_{j=0}^{n-1} f(x_j) = h \sum_{j=0}^{n-1} f(a + jh)$$

où $h = \frac{b-a}{n}$ et $x_j = a + jh$.

Erreur et Degré d'exactitude

L'erreur de la méthode des rectangles est d'ordre $O(h)$. Pour une fonction $f \in C^1([a, b])$, l'erreur $E_c(f) = \int_a^b f(t)dt - I_c(f)$ de la méthode des rectangles composite est majorée par :

$$|E_c(f)| \leq h \|f'\|_{\infty, [a, b]} (b - a) = \frac{(b - a)^2}{n} \|f'\|_{\infty, [a, b]}$$

Le degré d'exactitude de la méthode des rectangles élémentaire est 0, car elle est exacte pour les polynômes constants (de degré 0).

7.4.2 Méthode des Trapèzes

Définition

La méthode des trapèzes approche l'intégrale sur chaque sous-intervalle $[x_j, x_{j+1}]$ par l'aire du trapèze formé par les points $(x_j, 0)$, $(x_{j+1}, 0)$, $(x_{j+1}, f(x_{j+1}))$, $(x_j, f(x_j))$. Pour la méthode des trapèzes élémentaire sur un intervalle $[\alpha, \beta]$, on a:

$$\int_{\alpha}^{\beta} f(t)dt \approx \frac{(\beta - \alpha)}{2} [f(\alpha) + f(\beta)]$$

La formule de quadrature composite des trapèzes sur $[a, b]$ avec n intervalles est donnée par:

$$I_c(f) = \frac{h}{2} [f(x_0) + 2 \sum_{j=1}^{n-1} f(x_j) + f(x_n)] = \frac{h}{2} [f(a) + 2 \sum_{j=1}^{n-1} f(a + jh) + f(b)]$$

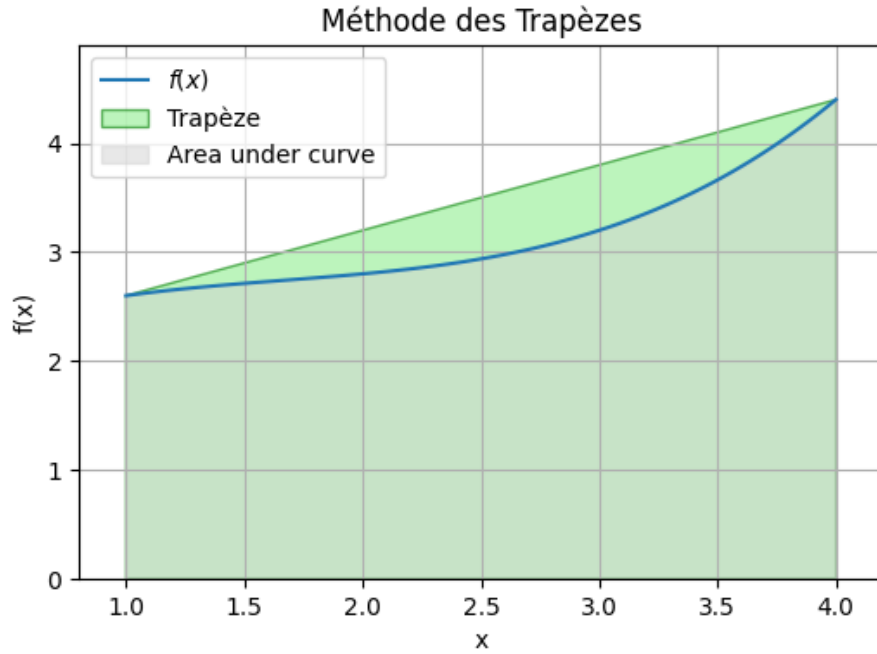


Figure 7.3: Méthode des Trapèzes.

où $h = \frac{b-a}{n}$ et $x_j = a + jh$.

Erreur et Degré d'exactitude

L'erreur de la méthode des trapèzes est d'ordre $O(h^2)$. Pour une fonction $f \in C^2([a, b])$, l'erreur $E_c(f) = \int_a^b f(t)dt - I_c(f)$ de la méthode des trapèzes composite est majorée par :

$$|E_c(f)| \leq \frac{h^2}{12} \|f''\|_{\infty, [a, b]} (b - a) = \frac{(b - a)^3}{12n^2} \|f''\|_{\infty, [a, b]}$$

Le degré d'exactitude de la méthode des trapèzes élémentaire est 1, car elle est exacte pour les polynômes de degré au plus 1.

7.4.3 Méthode du Point Milieu

Définition

La méthode du point milieu approche l'intégrale sur chaque sous-intervalle $[x_j, x_{j+1}]$ par l'aire du rectangle de hauteur $f(\frac{x_j + x_{j+1}}{2})$ au milieu de l'intervalle. Pour la méthode du point milieu élémentaire sur un intervalle $[\alpha, \beta]$, on a :

$$\int_{\alpha}^{\beta} f(t)dt \approx (\beta - \alpha) f\left(\frac{\alpha + \beta}{2}\right)$$

La formule de quadrature composite du point milieu sur $[a, b]$ avec n intervalles est donnée par :

$$I_c(f) = h \sum_{j=0}^{n-1} f\left(\frac{x_j + x_{j+1}}{2}\right) = h \sum_{j=0}^{n-1} f\left(a + \left(j + \frac{1}{2}\right)h\right)$$

où $h = \frac{b-a}{n}$ et $x_j = a + jh$.

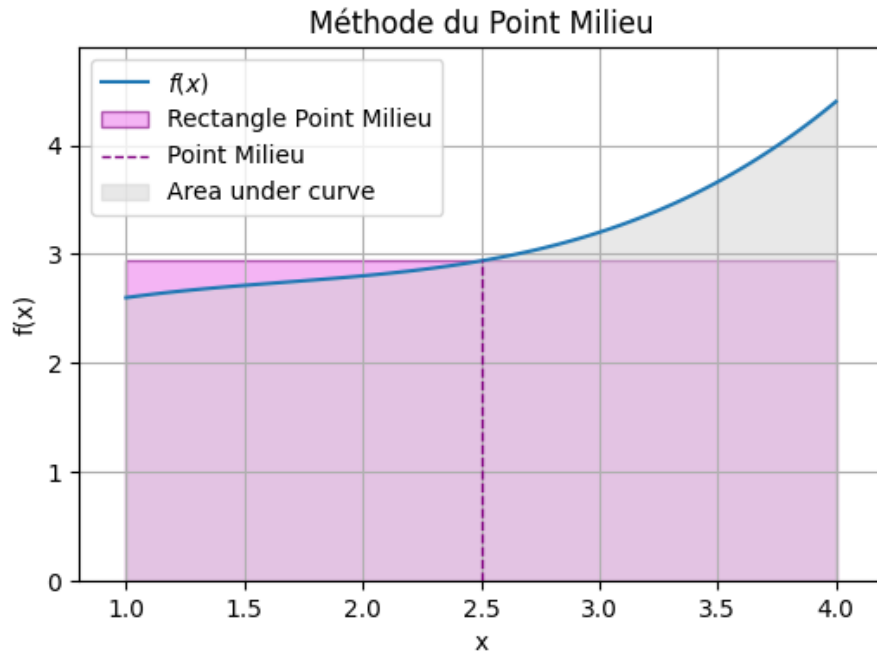


Figure 7.4: Méthode du Point Milieu.

Exercice : Degré d'exactitude

Déterminons le degré d'exactitude de la formule du point milieu élémentaire sur $[-1, 1]$. On teste pour les polynômes $f(x) = 1, x, x^2, \dots$

- Pour $f(x) = 1$:

$$\int_{-1}^1 1 dx = [x]_{-1}^1 = 2$$

$$I_e(f) = (1 - (-1))f\left(\frac{-1+1}{2}\right) = 2 \cdot f(0) = 2 \cdot 1 = 2$$

Donc, $I_e(f) = \int_{-1}^1 f(x) dx$.

- Pour $f(x) = x$:

$$\int_{-1}^1 x dx = \left[\frac{x^2}{2}\right]_{-1}^1 = 0$$

$$I_e(f) = 2 \cdot f(0) = 2 \cdot 0 = 0$$

Donc, $I_e(f) = \int_{-1}^1 f(x) dx$.

- Pour $f(x) = x^2$:

$$\int_{-1}^1 x^2 dx = \left[\frac{x^3}{3}\right]_{-1}^1 = \frac{2}{3}$$

$$I_e(f) = 2 \cdot f(0) = 2 \cdot 0^2 = 0$$

Donc, $I_e(f) \neq \int_{-1}^1 f(x) dx$.

- Pour $f(x) = x^3$:

$$\int_{-1}^1 x^3 dx = \left[\frac{x^4}{4} \right]_{-1}^1 = 0$$

$$I_e(f) = 2 \cdot f(0) = 2 \cdot 0^3 = 0$$

Donc, $I_e(f) = \int_{-1}^1 f(x) dx$.

- Pour $f(x) = x^4$:

$$\int_{-1}^1 x^4 dx = \left[\frac{x^5}{5} \right]_{-1}^1 = \frac{2}{5}$$

$$I_e(f) = 2 \cdot f(0) = 2 \cdot 0^4 = 0$$

Donc, $I_e(f) \neq \int_{-1}^1 f(x) dx$.

La formule élémentaire du point milieu est exacte pour les polynômes de degré au moins 1. En fait, elle est exacte pour les polynômes de degré au plus 1. Testons pour un polynôme de degré 2 pour être sûr. On a vu que pour $f(x) = x^2$, la formule n'est pas exacte. Testons pour $f(x) = s^2$:

$$\int_{-1}^1 s^2 ds = \frac{2}{3}$$

$$2 \cdot f(0) = 2 \cdot 0^2 = 0 \neq \frac{2}{3}$$

Donc la formule élémentaire n'est pas exacte pour les polynômes de degré 2.

****Conclusion :**** La formule du point milieu élémentaire est exacte pour les polynômes de degré 1. En fait, par un calcul plus précis et en considérant l'erreur, on montre que la formule du point milieu élémentaire est exacte pour les polynômes de degré au plus 1. On dit que le degré d'exactitude est 1. En réalité, le degré d'exactitude de la méthode du point milieu élémentaire est en fait 1, car elle est exacte pour les polynômes de degré au plus 1. La formule est d'ordre 2, donc l'erreur est en $O(h^2)$ pour les fonctions suffisamment régulières.

En réalité, la formule du point milieu élémentaire est exacte pour les polynômes de degré au plus 1. La formule du point milieu élémentaire est en fait exacte pour les polynômes de degré au plus 1. Donc le degré d'exactitude est 1. On dit que la formule du point milieu est d'ordre 2 car l'erreur est en $O(h^2)$.

7.4.4 Méthode de Simpson

Définition

La méthode de Simpson approche l'intégrale sur chaque intervalle double $[x_j, x_{j+2}]$ en utilisant un polynôme de degré 2 qui interpole les points $(x_j, f(x_j))$, $(x_{j+1}, f(x_{j+1}))$, $(x_{j+2}, f(x_{j+2}))$. Sur un intervalle $[\alpha, \beta]$, en posant $\gamma = \frac{\alpha+\beta}{2}$, la formule de Simpson élémentaire est:

$$\int_{\alpha}^{\beta} f(x) dx \approx \frac{(\beta - \alpha)}{6} \left[f(\alpha) + 4f\left(\frac{\alpha + \beta}{2}\right) + f(\beta) \right]$$

La formule de quadrature composite de Simpson sur $[a, b]$ avec n intervalles (n pair) est donnée par:

$$I_c(f) = \frac{h}{3} \left[f(x_0) + 4 \sum_{j=1}^{n/2} f(x_{2j-1}) + 2 \sum_{j=1}^{n/2-1} f(x_{2j}) + f(x_n) \right]$$

où $h = \frac{b-a}{n}$ et $x_j = a + jh$.

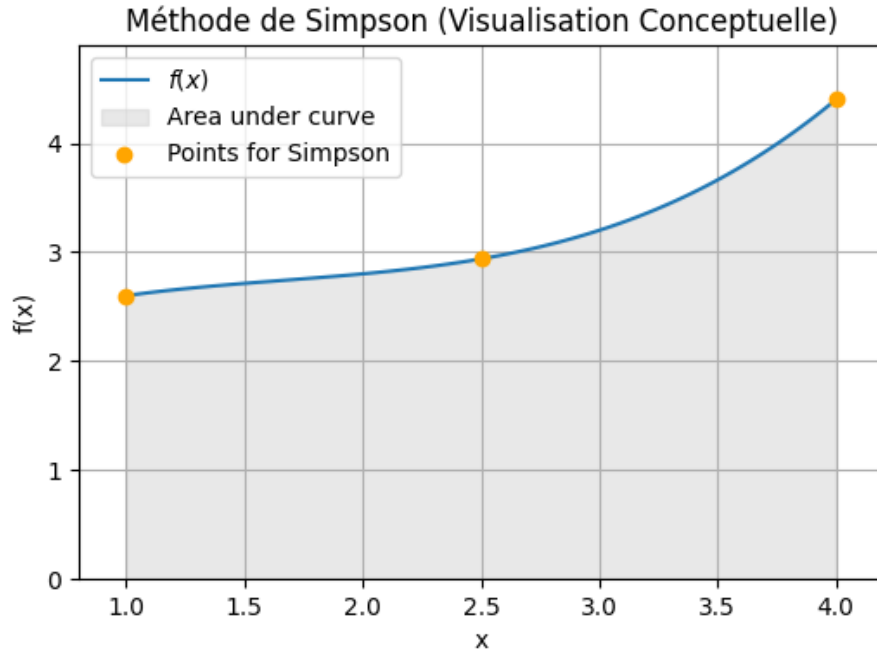


Figure 7.5: Méthode de Simpson (visualisation conceptuelle montrant l'aire sous la courbe).

Erreur et Degré d'exactitude

L'erreur de la méthode de Simpson est d'ordre $O(h^4)$. Pour une fonction $f \in C^4([a, b])$, l'erreur $E_c(f) = \int_a^b f(t)dt - I_c(f)$ de la méthode de Simpson composite est majorée par :

$$|E_c(f)| \leq \frac{h^4}{2880} \|f^{(4)}\|_{\infty, [a, b]} (b-a) = \frac{(b-a)^5}{2880n^4} \|f^{(4)}\|_{\infty, [a, b]}$$

Le degré d'exactitude de la méthode de Simpson élémentaire est 3, car elle est exacte pour les polynômes de degré au plus 3.

7.5 Comparaison des Méthodes

Proposition 7.5.1. Si $f \in C^2([a, b])$, l'erreur de quadrature de la méthode composite associée à une subdivision uniforme de pas h est majorée par :

$$|E_c(f)| = \left| \int_a^b f(t)dt - I_c(f) \right| \leq h^2 \frac{(b-a)}{12} \|f''\|_{\infty, [a, b]}$$

pour la méthode des trapèzes.

Proposition 7.5.2. Si $f \in C^4([a, b])$, l'erreur de quadrature de la méthode composite de Simpson associée à une subdivision uniforme de pas h est majorée par :

$$|E_c(f)| = \left| \int_a^b f(t)dt - I_c(f) \right| \leq h^4 \frac{(b-a)}{2880} \|f^{(4)}\|_{\infty, [a, b]}$$

Méthode	Formule élémentaire (sur $[\alpha, \beta]$)	Ordre de convergence	Degré d'exactitude
Rectangles (gauche)	$(\beta - \alpha)f(\alpha)$	$O(h)$	0
Trapèzes	$\frac{(\beta - \alpha)}{2}[f(\alpha) + f(\beta)]$	$O(h^2)$	1
Point Milieu	$(\beta - \alpha)f\left(\frac{\alpha + \beta}{2}\right)$	$O(h^2)$	1
Simpson	$\frac{(\beta - \alpha)}{6}\left[f(\alpha) + 4f\left(\frac{\alpha + \beta}{2}\right) + f(\beta)\right]$	$O(h^4)$	3

Table 7.1: Comparaison des méthodes de quadrature composites classiques.

7.6 Conclusion

Les formules de quadrature composites offrent des outils efficaces pour l'approximation numérique d'intégrales définies. Le choix de la méthode dépend de la régularité de la fonction à intégrer et de la précision souhaitée. La méthode de Simpson, avec son ordre de convergence élevé, est généralement préférée pour les fonctions régulières lorsque la précision est primordiale. Cependant, les méthodes des rectangles et des trapèzes peuvent être suffisantes dans des contextes où une précision moindre est acceptable ou pour des fonctions moins régulières.

7.7 Exercices

1. Calculer l'intégrale $\int_0^1 x^2 dx$ en utilisant les méthodes des rectangles gauches, des trapèzes et du point milieu avec $n = 4$ sous-intervalles. Comparer les résultats avec la valeur exacte de l'intégrale.
2. Estimer l'erreur pour chaque méthode utilisée dans l'exercice précédent en utilisant les bornes d'erreur théoriques.
3. Déterminer le nombre de sous-intervalles nécessaires pour que la méthode des trapèzes approche l'intégrale $\int_0^2 e^x dx$ avec une précision de 10^{-3} .
4. Calculer le degré d'exactitude de la formule de quadrature élémentaire des trapèzes sur $[-1, 1]$.

Chapter 8

CM8

8.1 Introduction aux Méthodes de Newton-Cotes

Les méthodes de Newton-Cotes sont une famille de formules de quadrature numérique (intégration numérique) basées sur l'évaluation de l'intégrande en des points équidistants. Elles constituent une généralisation des méthodes élémentaires d'intégration.

Definition 8.1.1 (Méthode de Newton-Cotes). On appelle méthode de Newton-Cotes d'ordre K la méthode élémentaire consistant à utiliser le polynôme d'interpolation d'ordre K , $P_K(x)$, associé aux $K+1$ points x_i équidistants :

$$x_i = \alpha + i \frac{\beta - \alpha}{K}, \quad i = 0, \dots, K$$

L'intégrale de la fonction $f(x)$ est alors approchée par l'intégrale de ce polynôme :

$$\int_{\alpha}^{\beta} f(x) dx \approx \int_{\alpha}^{\beta} P_K(x) dx = \sum_{i=0}^K \omega_i f(x_i)$$

où les poids ω_i sont donnés par :

$$\omega_i = \int_{\alpha}^{\beta} L_i(x) dx = \int_{\alpha}^{\beta} \prod_{\substack{j=0 \\ j \neq i}}^K \frac{x - x_j}{x_i - x_j} dx$$

$L_i(x)$ sont les polynômes de base de Lagrange.

8.2 Exemples de Formules et Cas Particuliers

L'ordre de la formule de Newton-Cotes est p . Cette formule est d'ordre K si K est impair, et d'ordre $K+1$ si K est pair. On n'utilise ces méthodes que pour K pair, sauf le cas $K=1$.

- Si $K=1$ (2 points), on a la formule des trapèzes.
- Si $K=2$ (3 points), on a la formule de Simpson.
- Si $K=4$ (5 points), on a la formule de Boole-Villarceau. Par exemple, pour l'intervalle $[0, 1]$:

$$\int_0^1 f(x) dx \approx \frac{1}{90} \left[7f(0) + 32f\left(\frac{1}{4}\right) + 12f\left(\frac{1}{2}\right) + 32f\left(\frac{3}{4}\right) + 7f(1) \right]$$

(Note: ceci correspond à $h = 1/4$, et les coefficients généraux sont $\frac{2h}{45}(7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4)$).

- Pour $K = 6$ (7 points), on a la formule de Hardy.

Pour $K \geq 8$, certains poids ω_i deviennent négatifs, ce qui rend les formules sensibles aux erreurs d'arrondi et peut entraîner une perte de précision.

8.3 Théorème et Erreur d'Intégration

Theorem 8.3.1. Soient $I(f) = \int_{\alpha}^{\beta} f(x)dx$ l'intégrale exacte et $I_K(f) = \sum_{i=0}^K \omega_i f(x_i)$ l'approximation par la méthode de Newton-Cotes. L'erreur d'intégration est $E(f) = I(f) - I_K(f)$. Supposons que la méthode d'intégration soit d'ordre $p \geq K$. Posons le noyau de Peano $K(t)$ comme suit :

$$K(t) = E_x((x-t)_+^p) = \int_{\alpha}^{\beta} (x-t)_+^p dx - \sum_{i=0}^K \omega_i (x_i - t)_+^p$$

où $(u)_+^p = u^p$ si $u > 0$ et 0 sinon. Alors, pour toute fonction $f \in C^{p+1}([\alpha, \beta])$, on a :

$$E(f) = \frac{1}{p!} \int_{\alpha}^{\beta} K(t) f^{(p+1)}(t) dt$$

Si $K(t)$ est de signe constant sur $[\alpha, \beta]$ (ce qui est le cas pour les méthodes de Newton-Cotes usuelles lorsque K n'est pas trop grand), alors il existe $c \in [\alpha, \beta]$ tel que :

$$E(f) = \frac{f^{(p+1)}(c)}{p!} \int_{\alpha}^{\beta} K(t) dt$$

On peut aussi écrire cela en utilisant l'erreur pour la fonction $x \mapsto x^{p+1}$:

$$E(f) = \frac{f^{(p+1)}(c)}{(p+1)!} E(x \mapsto x^{p+1})$$

En effet, $E(x \mapsto x^{p+1}) = \int_{\alpha}^{\beta} K(t) \frac{d^{p+1}(t^{p+1})}{dt^{p+1}} \frac{1}{p!} dt = \int_{\alpha}^{\beta} K(t) \frac{(p+1)!}{p!} dt = (p+1) \int_{\alpha}^{\beta} K(t) dt$. Dans les méthodes de Newton-Cotes (pour K petit), le noyau de Peano $K(t)$ a un signe constant.

8.4 Construction de Formules de Quadrature à Points Non Équidistants : Formule de Gauss-Legendre

On cherche s'il existe un meilleur choix des points x_1, \dots, x_N (non nécessairement équidistants) dans $[\alpha, \beta]$ et des poids $\omega_1, \dots, \omega_N$ pour que la formule de quadrature $\sum_{i=1}^N \omega_i f(x_i)$ soit exacte pour les polynômes de $\mathbb{R}_n[X]$ avec n le plus grand possible. Typiquement, pour N points, on peut atteindre un degré d'exactitude $2N - 1$.

Exemple 8.4.1 (Formule de Gauss-Legendre à 2 points). Cherchons une formule de la forme $\int_{-1}^1 f(x) dx \approx \omega_1 f(x_1) + \omega_2 f(x_2)$. Nous avons 4 inconnues $(x_1, x_2, \omega_1, \omega_2)$. Nous imposons donc que la formule soit

exacte pour les polynômes $1, x, x^2, x^3$.

$$\begin{aligned}\int_{-1}^1 1 \, dx &= 2 = \omega_1 + \omega_2 \\ \int_{-1}^1 x \, dx &= 0 = \omega_1 x_1 + \omega_2 x_2 \\ \int_{-1}^1 x^2 \, dx &= \frac{2}{3} = \omega_1 x_1^2 + \omega_2 x_2^2 \\ \int_{-1}^1 x^3 \, dx &= 0 = \omega_1 x_1^3 + \omega_2 x_2^3\end{aligned}$$

La résolution de ce système donne : $x_1 = -1/\sqrt{3}$, $x_2 = 1/\sqrt{3}$ (racines du polynôme de Legendre $P_2(x) = \frac{1}{2}(3x^2 - 1)$). $\omega_1 = 1$, $\omega_2 = 1$. La formule est donc :

$$\int_{-1}^1 f(x) dx \approx f(-1/\sqrt{3}) + f(1/\sqrt{3})$$

Cette formule est exacte pour les polynômes de degré ≤ 3 (car $2N - 1 = 2 \times 2 - 1 = 3$).

Exemple 8.4.2 (Formule de Gauss-Legendre à 3 points). On cherche $\int_{-1}^1 f(x) dx \approx \omega_1 f(x_1) + \omega_2 f(x_2) + \omega_3 f(x_3)$. On impose l'exactitude pour les polynômes $1, x, x^2, x^3, x^4, x^5$ (degré $2N - 1 = 2 \times 3 - 1 = 5$). On obtient que x_1, x_2, x_3 sont les racines du polynôme de Legendre de degré 3, $P_3(x) = \frac{1}{2}(5x^3 - 3x)$:

$$x_1 = -\sqrt{3/5}, \quad x_2 = 0, \quad x_3 = \sqrt{3/5}$$

Les poids sont :

$$\omega_1 = 5/9, \quad \omega_2 = 8/9, \quad \omega_3 = 5/9$$

La formule est :

$$\int_{-1}^1 f(x) dx \approx \frac{5}{9} f(-\sqrt{3/5}) + \frac{8}{9} f(0) + \frac{5}{9} f(\sqrt{3/5})$$

Proposition 8.4.3. Considérons la formule à N points $\int_{-1}^1 P(x) dx \approx \sum_{i=1}^N \omega_i P(x_i)$ qui est exacte pour les polynômes de degré $\leq 2N - 1$. Alors les abscisses x_1, \dots, x_N sont les N racines du polynôme de Legendre de degré N , noté $L_N(x)$ (ou $P_N(x)$), défini par la relation de récurrence : $L_0(x) = 1$, $L_1(x) = x$

$$L_n(x) = \frac{1}{n} [(2n-1)xL_{n-1}(x) - (n-1)L_{n-2}(x)] \quad \text{pour } n \geq 2$$

Les poids ω_i sont donnés par :

$$\omega_i = \int_{-1}^1 \prod_{\substack{j=1 \\ j \neq i}}^N \frac{x - x_j}{x_i - x_j} dx, \quad i = 1, \dots, N$$

La formule de quadrature ainsi obtenue est appelée Formule de Gauss-Legendre.

8.4.1 Méthode de résolution pratique (pour 2 points)

Pour retrouver la formule à 2 points $\int_{-1}^1 P(x) dx \approx \omega_1 f(x_1) + \omega_2 f(x_2)$: On considère le polynôme $\Pi_2(x) = (x - x_1)(x - x_2)$. Pour que la formule soit de Gauss, ce polynôme doit être orthogonal à tous les polynômes de degré inférieur à 2 sur $[-1, 1]$ avec un poids $w(x) = 1$.

1. Orthogonalité à $P_0(x) = 1$:

$$\begin{aligned}\int_{-1}^1 (x - x_1)(x - x_2)dx &= 0 \\ \int_{-1}^1 (x^2 - (x_1 + x_2)x + x_1x_2)dx &= 0 \\ \left[\frac{x^3}{3} - (x_1 + x_2)\frac{x^2}{2} + x_1x_2x \right]_{-1}^1 &= 0 \\ \frac{2}{3} + 2x_1x_2 &= 0 \Rightarrow x_1x_2 = -1/3\end{aligned}$$

2. Orthogonalité à $P_1(x) = x$:

$$\begin{aligned}\int_{-1}^1 x(x - x_1)(x - x_2)dx &= 0 \\ \int_{-1}^1 (x^3 - (x_1 + x_2)x^2 + x_1x_2x)dx &= 0 \\ \left[\frac{x^4}{4} - (x_1 + x_2)\frac{x^3}{3} + x_1x_2\frac{x^2}{2} \right]_{-1}^1 &= 0 \\ -\frac{2}{3}(x_1 + x_2) &= 0 \Rightarrow x_1 + x_2 = 0\end{aligned}$$

Comme $x_1 + x_2 = 0$ et $x_1x_2 = -1/3$, le polynôme $(x - x_1)(x - x_2)$ est $x^2 - (x_1 + x_2)x + x_1x_2 = x^2 - 1/3$. Les racines de $x^2 - 1/3 = 0$ sont $x^2 = 1/3 \Rightarrow x_1 = -1/\sqrt{3}, x_2 = 1/\sqrt{3}$.

Pour trouver les poids ω_1, ω_2 , on utilise l'exactitude de la formule pour des polynômes simples :

1. Pour $f(x) = 1$:

$$\int_{-1}^1 1dx = 2 = \omega_1 + \omega_2$$

2. Pour $f(x) = x$:

$$\int_{-1}^1 xdx = 0 = \omega_1(-1/\sqrt{3}) + \omega_2(1/\sqrt{3})$$

De la deuxième équation, $0 = (-\omega_1 + \omega_2)/\sqrt{3} \Rightarrow \omega_1 = \omega_2$. En substituant dans la première, $2 = \omega_1 + \omega_1 = 2\omega_1 \Rightarrow \omega_1 = 1$. Donc $\omega_1 = \omega_2 = 1$.

On retrouve bien la formule de Gauss-Legendre à 2 points.

Chapter 9

CM9

9.1 Solution approchée d'Équations Différentielles Ordinaires (EDO)

9.1.1 Motivations et Définitions

Notations et Définitions de base

Soit $F : [a, b] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ une fonction, telle que $(t, x) \mapsto F(t, x)$, avec $a, b \in \mathbb{R}$ et $d \in \mathbb{N}^*$. Cette fonction vectorielle F est donnée par ses composantes $F_i : [a, b] \times \mathbb{R}^d \rightarrow \mathbb{R}$ pour $i = 1, \dots, d$:

$$F(t, x) = \begin{pmatrix} F_1(t, x) \\ \vdots \\ F_d(t, x) \end{pmatrix}$$

On note $g^{(p)}$ la dérivée d'ordre p d'une fonction $g : \mathbb{R} \rightarrow \mathbb{R}$. En particulier, g' est la dérivée d'ordre 1.

Si une fonction $g : [a, b] \rightarrow \mathbb{R}^d$ est continue ainsi que toutes ses dérivées jusqu'à l'ordre p , on notera $g \in C^p([a, b], \mathbb{R}^d)$ ou simplement $g \in C^p([a, b])$ s'il n'y a pas d'ambiguïté sur l'espace d'arrivée. On a l'équivalence suivante :

$$(g_i \in C^k([a, b], \mathbb{R}) \quad \forall i = 1, \dots, d) \iff (g \in C^k([a, b], \mathbb{R}^d))$$

Définitions des EDOs

Definition 9.1.1. On appelle **équation différentielle d'ordre 1** une équation de la forme:

$$y'(t) = F(t, y(t)), \quad \forall t \in [t_0, t_0 + T]$$

où $y(t) \in \mathbb{R}^d$ et $F : [a, b] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$.

Definition 9.1.2. On appelle **EDO d'ordre p** une équation de la forme:

$$y^{(p)}(t) = f(t, y(t), y'(t), \dots, y^{(p-1)}(t))$$

où $f : [a, b] \times (\mathbb{R}^d)^p \rightarrow \mathbb{R}^d$ est une fonction continue.

Definition 9.1.3. Une fonction y de classe C^1 (ou C^p pour une EDO d'ordre p) vérifiant une EDO est dite **solution** de l'EDO. Résoudre une EDO, c'est déterminer toutes les solutions de cette EDO.

Lorsque $d \neq 1$ (c'est-à-dire $y(t)$ est un vecteur et non un scalaire), on parle de **système d'EDOs**.

Réduction à un système d'ordre 1 et Problème de Cauchy

Theorem 9.1.4. Toute EDO d'ordre $p \geq 1$ peut se ramener à un système d'EDOs d'ordre 1.

Preuve. Soit une EDO d'ordre p : $y^{(p)}(t) = f(t, y(t), y'(t), \dots, y^{(p-1)}(t))$. On pose $X_1(t) = y(t)$, $X_2(t) = y'(t)$, ..., $X_p(t) = y^{(p-1)}(t)$. Alors on a le système d'équations d'ordre 1 suivant pour le vecteur $X(t) = (X_1(t), \dots, X_p(t))^T$:

$$\begin{aligned} X_1'(t) &= y'(t) = X_2(t) \\ X_2'(t) &= y''(t) = X_3(t) \\ &\vdots \\ X_{p-1}'(t) &= y^{(p-1)}(t) = X_p(t) \\ X_p'(t) &= y^{(p)}(t) = f(t, X_1(t), X_2(t), \dots, X_p(t)) \end{aligned}$$

Ce qui peut s'écrire sous la forme $X'(t) = \mathcal{F}(t, X(t))$. □

Definition 9.1.5. On appelle **problème de Cauchy** pour une EDO d'ordre 1 la donnée de l'EDO et de la valeur de la solution en un point initial $t_0 \in [a, b]$. Le couple (t_0, y_0) , où $y_0 = y(t_0)$, est appelé **condition initiale**. Le problème de Cauchy consiste à résoudre :

$$\begin{cases} y'(t) = F(t, y(t)), & t \in [t_0, t_0 + T] \\ y(t_0) = y_0 \end{cases}$$

où $y_0 \in \mathbb{R}^d$ est donné.

La recherche d'une fonction de classe C^1 vérifiant ce système est l'objectif.

9.1.2 Exemples d'EDO

Pendule simple

L'équation du mouvement d'un pendule simple est $L\phi''(t) + g\sin(\phi(t)) = 0$, où $\phi(t)$ est l'angle par rapport à la verticale, L la longueur du pendule, et g l'accélération due à la gravité. Ceci est une EDO d'ordre 2 :

$$\phi''(t) + \frac{g}{L} \sin(\phi(t)) = 0$$

Pour la ramener à un système d'ordre 1, on pose $X_1(t) = \phi(t)$ et $X_2(t) = \phi'(t)$. Alors, le système devient :

$$\begin{aligned} X_1'(t) &= \phi'(t) = X_2(t) \\ X_2'(t) &= \phi''(t) = -\frac{g}{L} \sin(\phi(t)) = -\frac{g}{L} \sin(X_1(t)) \end{aligned}$$

Si on note $X(t) = \begin{pmatrix} X_1(t) \\ X_2(t) \end{pmatrix}$, alors $X'(t) = F(t, X(t))$ avec $F(t, X) = \begin{pmatrix} X_2 \\ -\frac{g}{L} \sin(X_1) \end{pmatrix}$. C'est un système d'EDOs d'ordre 1, où $F : \mathbb{R} \times \mathbb{R}^2 \rightarrow \mathbb{R}^2$.

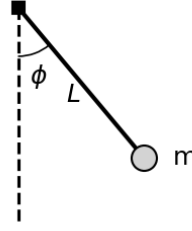


Figure 9.1: Schéma d'un pendule simple.

Chute libre avec frottement quadratique

L'équation du mouvement vertical $z(t)$ d'un objet en chute libre avec une résistance de l'air proportionnelle au carré de la vitesse $v(t) = z'(t)$ est donnée par:

$$z''(t) = -g + k(z)(z'(t))^2$$

où g est l'accélération de la pesanteur (constante) et $k(z)$ est un coefficient lié à la forme de l'objet et à la densité de l'air, pouvant dépendre de l'altitude z . C'est une EDO d'ordre 2.

On pose $Y(t) = \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} = \begin{pmatrix} z(t) \\ z'(t) \end{pmatrix}$. Le système d'ordre 1 équivalent est :

$$\begin{aligned} y_1'(t) &= z'(t) = y_2(t) \\ y_2'(t) &= z''(t) = -g + k(y_1(t))(y_2(t))^2 \end{aligned}$$

Soit $Y'(t) = F(t, Y(t))$ avec $F(t, Y) = \begin{pmatrix} Y_2 \\ -g + k(Y_1)Y_2^2 \end{pmatrix}$. C'est un système d'équations d'ordre 1. (Note: dans les notes manuscrites, il y a une formule $F(t, Y) = \begin{pmatrix} V \\ -g + ky_1y_2^2 \end{pmatrix}$ qui semble contenir une coquille si $Y = (y_1, y_2) = (z, V)$, le terme $ky_1y_2^2$ devrait être $k(y_1)y_2^2$).

Modèle épidémiologique SIR

Un modèle simple d'épidémie divise la population en trois compartiments: Susceptibles (S), Infectés (I), et Rétablis (R). Les transitions sont: $S \xrightarrow{k_1 I S} I \xrightarrow{k_2 I} R$. k_1 est le taux d'infection, k_2 est le taux de guérison. Le système d'EDOs est:

$$\begin{aligned} S'(t) &= -k_1 I(t)S(t) \\ I'(t) &= k_1 I(t)S(t) - k_2 I(t) \\ R'(t) &= k_2 I(t) \end{aligned}$$

C'est un système d'EDOs d'ordre 1. $Y(t) = (S(t), I(t), R(t))^T$.

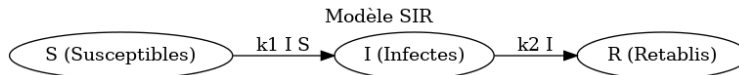


Figure 9.2: Diagramme du modèle épidémiologique SIR.

Exemple de fonction $F(t, Y)$ pour un système

La fonction $F(t, Y)$ pour un système $Y'(t) = F(t, Y(t))$ peut prendre diverses formes. Par exemple, considérons $Y = (y_1, y_2)^T$. Une fonction F pourrait être:

Listing 9.1: Exemple de fonction $F(t$

```
import numpy as np

# g, L, k sont des parametres supposes definis ailleurs
# Par exemple: g = 9.81, L = 1.0, k = 0.1

def F(t, Y):
    y1, y2 = Y # Y est un vecteur [y1, y2]
    dY_dt = np.array([
        y2,
        -g/L * k * y1 * y2**2
    ]) * np.exp(-t)
    return dY_dt
```

Ce code définit le système d'EDO:

$$\begin{aligned} y_1'(t) &= y_2(t)e^{-t} \\ y_2'(t) &= \left(-\frac{g}{L}ky_1(t)y_2(t)^2\right)e^{-t} \end{aligned}$$

9.1.3 Problème de Cauchy et Existence/Unicité de la solution

Nécessité de la solution approchée

On considère le problème de Cauchy suivant:

1. $y'(t) = f(t, y(t))$, pour $t \in [t_0, t_0 + T]$
2. $y(t_0) = y_0$ (condition initiale donnée)

Souvent, on ne sait pas résoudre analytiquement ce système (1)-(2), sauf dans des cas particuliers (par exemple, EDO linéaires à coefficients constants, EDO à variables séparables, etc.).

Exemple 9.1.6. L'EDO $y'(t) = \sin(t^2 y(t))$ avec $y(0) = 1$ est un exemple d'équation dont on ne connaît pas de solution analytique explicite.

D'où la nécessité de recourir à des méthodes de solution approchée.

Théorème de Cauchy-Lipschitz

Sous certaines conditions sur la fonction f , le problème de Cauchy admet une solution unique.

Définition 9.1.7 (Fonction Lipschitzienne). On dit qu'une fonction $f : D \subset [a, b] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ est **Lipschitzienne** par rapport à sa seconde variable sur D s'il existe une constante $L > 0$ (appelée constante de Lipschitz) telle que pour tous $(t, y_1) \in D$ et $(t, y_2) \in D$:

$$\|f(t, y_1) - f(t, y_2)\|_{\mathbb{R}^d} \leq L\|y_1 - y_2\|_{\mathbb{R}^d}$$

Theorem 9.1.8 (Cauchy-Lipschitz). Soit le problème de Cauchy (P) : $y'(t) = f(t, y(t))$ avec $y(t_0) = y_0$. Si $f : [t_0, t_0 + T] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ est une fonction telle que:

1. f est continue sur $[t_0, t_0 + T] \times \mathbb{R}^d$.
2. f est Lipschitzienne par rapport à sa seconde variable y sur $[t_0, t_0 + T] \times \mathbb{R}^d$.

Alors, le problème de Cauchy (P) admet une unique solution $y(t)$ de classe $C^1([t_0, t_0 + T], \mathbb{R}^d)$.

Qualités de la solution théorique

Lorsque le problème est bien posé (c'est-à-dire que les conditions du théorème de Cauchy-Lipschitz sont satisfaites), la solution théorique possède certaines qualités:

- Existence et unicité de la solution.
- Régularité de la solution (par exemple, C^1).
- Dépendance continue de la solution vis-à-vis des données du problème (condition initiale y_0 et fonction f). Ceci est lié à la stabilité du problème.

Cependant, même si le problème est bien posé et admet une solution unique, on peut ne pas être capable de la trouver analytiquement.

9.1.4 Schémas Numériques à un pas

Formulation Intégrale du problème de Cauchy

L'EDO $y'(t) = f(t, y(t))$ peut être intégrée entre t_0 et t :

$$\int_{t_0}^t y'(s) ds = \int_{t_0}^t f(s, y(s)) ds$$
$$y(t) - y(t_0) = \int_{t_0}^t f(s, y(s)) ds$$

Proposition 9.1.9. Une fonction $y(t)$ est solution du problème de Cauchy (P)

$$\begin{cases} y'(t) = f(t, y(t)) \\ y(t_0) = y_0 \end{cases}$$

si et seulement si $y(t)$ satisfait l'équation intégrale:

$$y(t) = y_0 + \int_{t_0}^t f(s, y(s)) ds, \quad \forall t \in [t_0, t_0 + T]$$

Cette formulation est la base de nombreuses méthodes numériques.

Principe des méthodes numériques et maillage temporel

L'idée est de construire une suite de points (t_n, y_n) qui approximent la solution $y(t)$ aux instants t_n . On définit un **maillage** du temps: une suite discrète d'instant $t_n = t_0 + n\Delta t$ pour $n = 0, 1, \dots, N$, où $\Delta t = T/N$ est le **pas de temps** et N est le nombre de pas. On calcule $y_n \approx y(t_n)$ pour chaque n . En reliant les points (t_n, y_n) , on obtient une approximation graphique de la solution $t \mapsto y(t)$.

Construction des schémas à partir d'approximations d'intégrales (Formules de Quadrature)

Les schémas numériques sont souvent dérivés de la formulation intégrale sur un petit intervalle $[t_n, t_{n+1}]$:

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

L'intégrale $\int_{t_n}^{t_{n+1}} g(s) ds$ (où $g(s) = f(s, y(s))$) est approximée par une formule de quadrature:

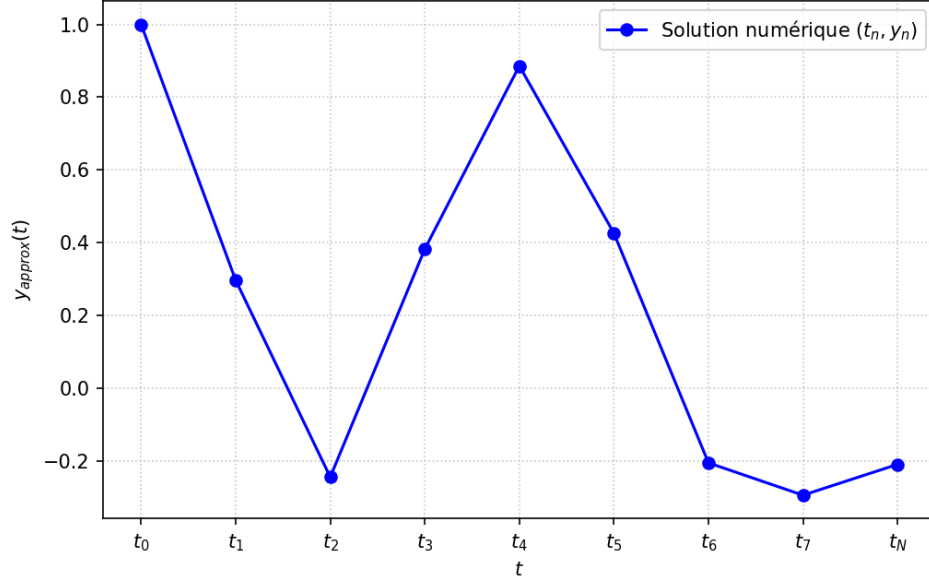


Figure 9.3: Illustration de la construction d'une solution numérique approchée $y_n \approx y(t_n)$. Les points (t_n, y_n) sont reliés pour former une approximation de la courbe solution.

- **Rectangle à gauche:** $\int_{t_n}^{t_{n+1}} g(s)ds \approx \Delta t \cdot g(t_n)$. Erreur $O(\Delta t^2)$.
- **Rectangle à droite:** $\int_{t_n}^{t_{n+1}} g(s)ds \approx \Delta t \cdot g(t_{n+1})$. Erreur $O(\Delta t^2)$.
- **Trapèze:** $\int_{t_n}^{t_{n+1}} g(s)ds \approx \frac{\Delta t}{2}[g(t_n) + g(t_{n+1})]$. Erreur $O(\Delta t^3)$.
- **Point milieu:** $\int_{t_n}^{t_{n+1}} g(s)ds \approx \Delta t \cdot g(t_n + \frac{\Delta t}{2})$. Erreur $O(\Delta t^3)$.

Sur l'intervalle $[t_n, t_{n+1}]$, on considère le problème de Cauchy local:

$$\begin{cases} y'(t) = f(t, y(t)) \\ y(t_n) = x_n \end{cases}$$

où x_n est la valeur approchée de $y(t_n)$ calculée à l'étape précédente ($x_0 = y_0$ étant la condition initiale exacte). Alors $y(t_{n+1}) = x_n + \int_{t_n}^{t_{n+1}} f(s, y(s))ds$. En approximant l'intégrale et $y(s)$ dans l'intégrande, on obtient différents schémas. On pose x_{n+1} comme l'approximation de $y(t_{n+1})$.

Schéma d'Euler Explicite

En utilisant la formule du rectangle à gauche pour l'intégrale et en approximant $f(s, y(s)) \approx f(t_n, x_n)$ pour $s \in [t_n, t_{n+1}]$:

$$y(t_{n+1}) \approx x_n + \Delta t f(t_n, x_n)$$

Le schéma d'Euler explicite est défini par:

$$x_{n+1} = x_n + \Delta t f(t_n, x_n), \quad n = 0, \dots, N-1$$

$$x_0 = y_0$$

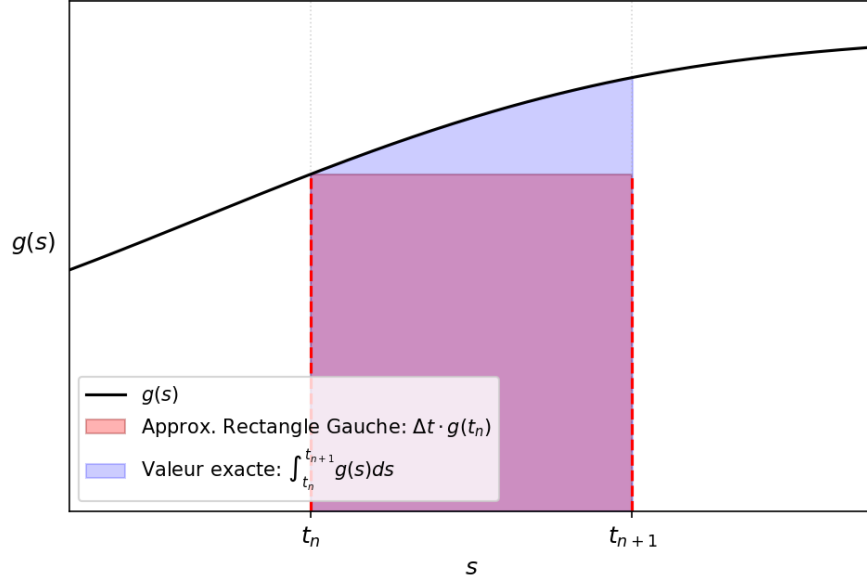


Figure 9.4: Approximation de l'intégrale $\int_{t_n}^{t_{n+1}} g(s)ds$ par la méthode du rectangle à gauche.

Schéma du Point Milieu (Explicite)

En utilisant la formule du point milieu pour l'intégrale:

$$\int_{t_n}^{t_{n+1}} f(s, y(s))ds \approx \Delta t f\left(t_n + \frac{\Delta t}{2}, y\left(t_n + \frac{\Delta t}{2}\right)\right)$$

Pour évaluer $y(t_n + \frac{\Delta t}{2})$, on peut utiliser une approximation d'Euler sur un demi-pas: $y(t_n + \frac{\Delta t}{2}) \approx y(t_n) + \frac{\Delta t}{2} f(t_n, y(t_n))$. En substituant $y(t_n)$ par x_n , on obtient le schéma du point milieu (un schéma de Runge-Kutta d'ordre 2):

$$x_{n+1} = x_n + \Delta t f\left(t_n + \frac{\Delta t}{2}, x_n + \frac{\Delta t}{2} f(t_n, x_n)\right), \quad n = 0, \dots, N-1$$

$$x_0 = y_0$$

La note indique $y(t_{n+1}) = y(t_n) + \Delta t f(t_n + \frac{\Delta t}{2}, y(t_n + \frac{\Delta t}{2})) + O(\Delta t^3)$. Ce $O(\Delta t^3)$ est l'erreur de troncature locale.

Schéma d'Euler Implicite

En utilisant la formule du rectangle à droite pour l'intégrale:

$$\int_{t_n}^{t_{n+1}} f(s, y(s))ds \approx \Delta t f(t_{n+1}, y(t_{n+1}))$$

On obtient:

$$y(t_{n+1}) \approx x_n + \Delta t f(t_{n+1}, y(t_{n+1}))$$

Le schéma d'Euler implicite est défini par:

$$x_{n+1} = x_n + \Delta t f(t_{n+1}, x_{n+1}), \quad n = 0, \dots, N-1$$

$$x_0 = y_0$$

Ce schéma est implicite car x_{n+1} apparaît des deux côtés de l'équation. À chaque pas de temps, on doit résoudre l'équation (souvent non linéaire) (PH): $x_{n+1} - \Delta t f(t_{n+1}, x_{n+1}) = x_n$ pour trouver x_{n+1} .

Schéma de Crank-Nicolson (Trapèze Implicite)

En utilisant la formule du trapèze pour l'intégrale:

$$\int_{t_n}^{t_{n+1}} f(s, y(s)) ds \approx \frac{\Delta t}{2} [f(t_n, y(t_n)) + f(t_{n+1}, y(t_{n+1}))]$$

Le schéma de Crank-Nicolson est défini par:

$$x_{n+1} = x_n + \frac{\Delta t}{2} [f(t_n, x_n) + f(t_{n+1}, x_{n+1})], \quad n = 0, \dots, N-1$$
$$x_0 = y_0$$

Ce schéma est également implicite.

Schéma de Heun (Trapèze Explicite / Euler amélioré)

On peut rendre le schéma du trapèze explicite en utilisant une prédiction pour x_{n+1} dans le terme $f(t_{n+1}, x_{n+1})$. Une méthode courante est d'utiliser une prédiction par Euler explicite:

1. Prédiction (Euler explicite): $\tilde{x}_{n+1} = x_n + \Delta t f(t_n, x_n)$
2. Correction (Trapèze): $x_{n+1} = x_n + \frac{\Delta t}{2} [f(t_n, x_n) + f(t_{n+1}, \tilde{x}_{n+1})]$

Ceci est le schéma de Heun:

$$x_{n+1} = x_n + \frac{\Delta t}{2} [f(t_n, x_n) + f(t_{n+1}, x_n + \Delta t f(t_n, x_n))], \quad n = 0, \dots, N-1$$
$$x_0 = y_0$$

C'est un schéma explicite.

Forme générale des schémas explicites à un pas

De nombreux schémas explicites à un pas peuvent s'écrire sous la forme générale:

$$x_{n+1} = x_n + \Delta t \Phi(t_n, x_n, \Delta t), \quad n = 0, \dots, N-1$$
$$x_0 \text{ donné}$$

où $\Phi : \mathbb{R} \times \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}^d$ est la fonction d'incrément.

Exemple 9.1.10. Exemples de fonctions Φ :

- **Euler explicite:** $\Phi(t, y, \Delta t) = f(t, y)$
- **Point Milieu:** $\Phi(t, y, \Delta t) = f\left(t + \frac{\Delta t}{2}, y + \frac{\Delta t}{2} f(t, y)\right)$
- **Heun:** $\Phi(t, y, \Delta t) = \frac{1}{2} [f(t, y) + f(t + \Delta t, y + \Delta t f(t, y))]$

9.1.5 Exemple: Modèle Proie-Prédateur (Lotka-Volterra)

Un autre exemple classique de système d'EDO est le modèle de Lotka-Volterra, qui décrit la dynamique des populations de proies $L(t)$ (par exemple, lapins) et de prédateurs $R(t)$ (par exemple, renards). On a le problème de Cauchy:

$$L'(t) = aL(t) - bL(t)R(t)$$
$$R'(t) = -cR(t) + dL(t)R(t)$$

avec les conditions initiales $L(0) = L_0$ et $R(0) = R_0$. Les constantes a, b, c, d sont positives et décrivent les interactions:

- $aL(t)$: croissance exponentielle des proies en l'absence de prédateurs.
- $-bL(t)R(t)$: mortalité des proies due à la prédation.
- $-cR(t)$: mortalité exponentielle des prédateurs en l'absence de proies.
- $dL(t)R(t)$: croissance des prédateurs grâce à la prédation.

Les notes mentionnent $L'(t) = L(t) - L(t)R(t)$ et $R'(t) = -R(t) + L(t)R(t)$, ce qui correspond au cas $a = 1, b = 1, c = 1, d = 1$. Ce système n'est généralement pas résoluble analytiquement et nécessite des méthodes numériques.

Chapter 10

CM10

10.1 Schémas numériques pour Équations Différentielles Ordinaires

10.1.1 Le Problème de Cauchy

On s'intéresse à la résolution numérique du problème de Cauchy (P) suivant :

$$(P) \quad \begin{cases} x'(t) = f(t, x(t)), & t \in [t_0, t_0 + T] \\ x(t_0) = x_0 \end{cases}$$

où $f : [t_0, t_0 + T] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ est une fonction donnée, $x_0 \in \mathbb{R}^d$ est la condition initiale, et $x(t) \in \mathbb{R}^d$ est la solution recherchée.

10.1.2 Schémas à un pas explicites

Pour approcher la solution de (P), on construit des schémas numériques. Un schéma à un pas explicite (S) se présente sous la forme générale :

$$(S) \quad \begin{cases} x_{n+1} = x_n + \Delta t \Phi(t_n, x_n, \Delta t) \\ x_0 \text{ donné (souvent } x_0 = x(t_0)) \end{cases}$$

où :

- $t_n = t_0 + n\Delta t$ sont les points de la discrétisation en temps, pour $n = 0, \dots, N$.
- $\Delta t = T/N$ est le pas de temps, avec N le nombre total de pas.
- x_n est une approximation de la solution $x(t_n)$.
- Φ est la fonction d'incrément qui caractérise le schéma.

10.1.3 Exemples de schémas à un pas explicites

Schéma d'Euler explicite

Pour ce schéma, la fonction d'incrément est $\Phi(t_n, x_n, \Delta t) = f(t_n, x_n)$. Le schéma s'écrit :

$$x_{n+1} = x_n + \Delta t f(t_n, x_n)$$

Schéma du Point Milieu (Runge-Kutta d'ordre 2)

La fonction d'incrément est $\Phi(t_n, x_n, \Delta t) = f\left(t_n + \frac{\Delta t}{2}, x_n + \frac{\Delta t}{2} f(t_n, x_n)\right)$. Le schéma s'écrit :

$$x_{n+1} = x_n + \Delta t f\left(t_n + \frac{\Delta t}{2}, x_n + \frac{\Delta t}{2} f(t_n, x_n)\right)$$

Schéma de Heun (Runge-Kutta d'ordre 2)

La fonction d'incrément est $\Phi(t_n, x_n, \Delta t) = \frac{1}{2} [f(t_n, x_n) + f(t_n + \Delta t, x_n + \Delta t f(t_n, x_n))]$. Le schéma s'écrit :

$$x_{n+1} = x_n + \frac{\Delta t}{2} [f(t_n, x_n) + f(t_n + \Delta t, x_n + \Delta t f(t_n, x_n))]$$

10.2 Étude de convergence pour les EDO

On s'intéresse à savoir si la suite $(x_n)_{0 \leq n \leq N}$ générée par un schéma (S) converge vers la solution $x(t)$ du problème (P) lorsque $\Delta t \rightarrow 0$.

10.2.1 Convergence

Definition 10.2.1 (Convergence d'un schéma). Le schéma (S) est dit convergent si, pour toute solution x de (P) et pour toute suite $(x_n)_{0 \leq n \leq N}$ construite par (S) avec $x_0 = x(t_0)$, on a :

$$\lim_{\Delta t \rightarrow 0} \left(\max_{0 \leq n \leq N} \|x_n - x(t_n)\| \right) = 0$$

Remark 10.2.2. Si la suite $(x_n)_{0 \leq n \leq N}$ est "bien" construite (par exemple, si le problème (P) admet une solution unique x et si f est de classe C^p assurant une régularité suffisante à x), la convergence peut être établie. L'erreur globale $\max_{0 \leq n \leq N} \|x_n - x(t_n)\|$ dépend de Δt . En général, la convergence d'un schéma numérique est la conséquence de sa consistance et de sa stabilité (Théorème de Lax-Richtmyer, adapté aux EDO).

10.2.2 Consistance et Ordre d'un schéma

Remark 10.2.3. Pour analyser un schéma numérique pour EDO, on procède typiquement en deux étapes :

1. On étudie la consistance du schéma (quelle est l'erreur commise en un seul pas ?).
2. On examine sa stabilité (comment les erreurs se propagent-elles ?).

Ces deux propriétés permettent ensuite de conclure sur la convergence.

Erreur de consistance (Erreur locale de troncature)

Definition 10.2.4 (Erreur de consistance). Soit $x(t)$ la solution exacte du problème (P). On appelle erreur locale de troncature (ou erreur de consistance) du schéma (S) à l'instant t_{n+1} , la quantité e_{n+1} définie par :

$$e_{n+1} = x(t_{n+1}) - (x(t_n) + \Delta t \Phi(t_n, x(t_n), \Delta t))$$

Cette erreur mesure à quel point la solution exacte $x(t)$ échoue à satisfaire l'équation du schéma numérique.

Ordre d'un schéma

Definition 10.2.5 (Ordre d'un schéma). Le schéma (S) est dit consistant d'ordre (au moins) $q \geq 1$ si, pour toute solution exacte $x(t)$ de (P) suffisamment régulière, il existe une constante $C > 0$ (indépendante de Δt) telle que :

$$\max_{0 \leq n \leq N-1} \|e_{n+1}\| \leq C(\Delta t)^{q+1}$$

Autrement dit, $e_{n+1}(\Delta t) = O((\Delta t)^{q+1})$. Si un schéma est consistant, alors $q \geq 1$. Le schéma est dit d'ordre q s'il est d'ordre au moins q et pas d'ordre au moins $q + 1$. Cela signifie que $e_{n+1}(\Delta t) = K(\Delta t)^{q+1} + O((\Delta t)^{q+2})$ avec $K \neq 0$.

Exemple 10.2.6 (Étude de consistance du schéma d'Euler explicite). Pour le schéma d'Euler explicite, $\Phi(t, y, \Delta t) = f(t, y)$. L'erreur locale de troncature est, pour une solution $x(t)$ suffisamment régulière :

$$e(t, \Delta t) = x(t + \Delta t) - (x(t) + \Delta t f(t, x(t)))$$

Comme $x(t)$ est solution de (P), on a $x'(t) = f(t, x(t))$. Donc :

$$e(t, \Delta t) = x(t + \Delta t) - x(t) - \Delta t x'(t)$$

En effectuant un développement de Taylor de $x(t + \Delta t)$ autour de t , en supposant $x \in C^2([t_0, t_0 + T])$:

$$x(t + \Delta t) = x(t) + \Delta t x'(t) + \frac{(\Delta t)^2}{2} x''(t) + O((\Delta t)^3)$$

En substituant ce développement dans l'expression de $e(t, \Delta t)$:

$$\begin{aligned} e(t, \Delta t) &= \left(x(t) + \Delta t x'(t) + \frac{(\Delta t)^2}{2} x''(t) + O((\Delta t)^3) \right) - x(t) - \Delta t x'(t) \\ &= \frac{(\Delta t)^2}{2} x''(t) + O((\Delta t)^3) \end{aligned}$$

Ainsi, pour le schéma d'Euler explicite, l'erreur locale de troncature à l'instant t_n est $e_{n+1} = \frac{(\Delta t)^2}{2} x''(t_n) + O((\Delta t)^3)$. On a $q + 1 = 2$, donc $q = 1$. Le schéma d'Euler explicite est d'ordre (au moins) 1 si $x \in C^2$. Si de plus $x \in C^3$ et $x''(t_n) \neq 0$ pour au moins un t_n , alors le schéma est exactement d'ordre 1.

Conditions d'ordre pour une fonction d'incrément Φ

Remark 10.2.7 (Régularité de Φ). Soit $\Phi : I \times \mathbb{R}^d \times [0, \Delta t_0] \rightarrow \mathbb{R}^d$, où $I = [t_0, t_0 + T]$ et $\Delta t_0 > 0$. La fonction Φ est dite de classe C^p si toutes ses dérivées partielles par rapport à t , x (ou y), et Δt jusqu'à l'ordre p existent et sont continues sur $I \times \mathbb{R}^d \times [0, \Delta t_0]$.

Pour une fonction d'incrément Φ de classe C^q , le schéma $x_{n+1} = x_n + \Delta t \Phi(t_n, x_n, \Delta t)$ est consistant d'ordre au moins q si et seulement si les conditions suivantes sont vérifiées pour tout (t, y) dans le domaine de définition :

1. $\Phi(t, y, 0) = f(t, y)$ (Condition pour l'ordre 0, implique $q \geq 1$ pour la consistance)
2. $\frac{\partial^k \Phi}{\partial (\Delta t)^k}(t, y, 0) = \frac{1}{k+1} D_t^k f(t, y)$, pour $k = 1, \dots, q-1$.

où $D_t f(t, y)$ est l'opérateur de dérivée totale par rapport à t le long des solutions de l'EDO:

$$D_t f(t, y) = \frac{\partial f}{\partial t}(t, y) + \frac{\partial f}{\partial y}(t, y) \cdot f(t, y)$$

et $D_t^0 f = f$, $D_t^k f = D_t(D_t^{k-1} f)$ pour $k \geq 1$.

Exemple 10.2.8 (Étude de consistance du schéma de Runge-Kutta d'ordre 2 (Point Milieu)). Le schéma est $x_{n+1} = x_n + \Delta t f(t_n + \frac{\Delta t}{2}, x_n + \frac{\Delta t}{2} f(t_n, x_n))$. La fonction d'incrément est $\Phi(t, y, \Delta t) = f(t + \frac{\Delta t}{2}, y + \frac{\Delta t}{2} f(t, y))$. Vérifions les conditions d'ordre :

1. Condition pour $q \geq 1$:

$$\Phi(t, y, 0) = f(t + 0, y + 0 \cdot f(t, y)) = f(t, y)$$

Cette condition est vérifiée. Donc le schéma est au moins d'ordre 1.

2. Condition pour $q \geq 2$ (correspond à $k = 1$ dans la formule générale): On doit vérifier si $\frac{\partial \Phi}{\partial(\Delta t)}(t, y, 0) = \frac{1}{2} D_t f(t, y)$. Calculons $\frac{\partial \Phi}{\partial(\Delta t)}$: Soient $u_1(t, \Delta t) = t + \frac{\Delta t}{2}$ et $u_2(y, \Delta t) = y + \frac{\Delta t}{2} f(t, y)$. Alors $\Phi(t, y, \Delta t) = f(u_1, u_2)$.

$$\begin{aligned} \frac{\partial \Phi}{\partial(\Delta t)}(t, y, \Delta t) &= \frac{\partial f}{\partial u_1}(u_1, u_2) \cdot \frac{\partial u_1}{\partial(\Delta t)} + \frac{\partial f}{\partial u_2}(u_1, u_2) \cdot \frac{\partial u_2}{\partial(\Delta t)} \\ &= \frac{\partial f}{\partial t} \left(t + \frac{\Delta t}{2}, y + \frac{\Delta t}{2} f(t, y) \right) \cdot \frac{1}{2} + \frac{\partial f}{\partial y} \left(t + \frac{\Delta t}{2}, y + \frac{\Delta t}{2} f(t, y) \right) \cdot \left(\frac{1}{2} f(t, y) \right) \end{aligned}$$

Évaluons en $\Delta t = 0$:

$$\begin{aligned} \frac{\partial \Phi}{\partial(\Delta t)}(t, y, 0) &= \frac{1}{2} \frac{\partial f}{\partial t}(t, y) + \frac{1}{2} \frac{\partial f}{\partial y}(t, y) f(t, y) \\ &= \frac{1}{2} \left(\frac{\partial f}{\partial t}(t, y) + \frac{\partial f}{\partial y}(t, y) f(t, y) \right) \\ &= \frac{1}{2} D_t f(t, y) \end{aligned}$$

Cette condition est vérifiée. Donc le schéma est au moins d'ordre 2.

3. Condition pour $q \geq 3$ (correspond à $k = 2$ dans la formule générale): On devrait vérifier si $\frac{\partial^2 \Phi}{\partial(\Delta t)^2}(t, y, 0) = \frac{1}{3} D_t^2 f(t, y)$. En général, cette condition n'est pas satisfaite pour une fonction f arbitraire. Pour montrer que le schéma n'est pas d'ordre au moins 3 (et donc qu'il est exactement d'ordre 2), il faudrait calculer $\frac{\partial^2 \Phi}{\partial(\Delta t)^2}(t, y, 0)$ et montrer que cette expression est différente de $\frac{1}{3} D_t^2 f(t, y)$, ou que le terme $g(t, y)$ dans $e_{n+1} = g(t, y)(\Delta t)^3 + O((\Delta t)^4)$ est non nul.

Le schéma du point milieu est donc d'ordre 2.

10.2.3 Stabilité

Définition 10.2.9 (Stabilité d'un schéma). Le schéma (S) est dit stable pour une classe de fonctions f s'il existe une constante $S > 0$, indépendante de Δt , telle que pour toutes suites $(x_n)_{0 \leq n \leq N}$ et $(y_n)_{0 \leq n \leq N}$ vérifiant :

$$\begin{aligned} x_{n+1} &= x_n + \Delta t \Phi(t_n, x_n, \Delta t) \\ y_{n+1} &= y_n + \Delta t \Phi(t_n, y_n, \Delta t) + \delta_n \quad (\text{perturbation}) \end{aligned}$$

pour $n = 0, \dots, N-1$, où les δ_n sont des perturbations, on ait la majoration :

$$\max_{0 \leq n \leq N} \|x_n - y_n\| \leq S \left(\|x_0 - y_0\| + \sum_{j=0}^{N-1} \|\delta_j\| \right)$$

La stabilité garantit que de petites perturbations (erreurs initiales ou erreurs introduites à chaque pas)

n'entraînent pas une divergence incontrôlée des solutions numériques.

Proposition 10.2.10 (Stabilité du schéma d'Euler explicite). Si la fonction $f(t, y)$ est Lipschitzienne par rapport à y , uniformément en t , c'est-à-dire s'il existe $L_f > 0$ telle que $\|f(t, y_1) - f(t, y_2)\| \leq L_f \|y_1 - y_2\|$ pour tous t, y_1, y_2 , alors le schéma d'Euler explicite est stable.

Preuve. Soient les suites (x_n) et (y_n) définies par :

$$\begin{aligned} x_{n+1} &= x_n + \Delta t f(t_n, x_n) \\ y_{n+1} &= y_n + \Delta t f(t_n, y_n) + \delta_n \end{aligned}$$

Soustrayons les deux équations :

$$x_{n+1} - y_{n+1} = (x_n - y_n) + \Delta t (f(t_n, x_n) - f(t_n, y_n)) - \delta_n$$

En prenant la norme et en utilisant l'inégalité triangulaire et la condition de Lipschitz sur f :

$$\begin{aligned} \|x_{n+1} - y_{n+1}\| &\leq \|x_n - y_n + \Delta t (f(t_n, x_n) - f(t_n, y_n))\| + \|\delta_n\| \\ &\leq \|x_n - y_n\| + \Delta t \|f(t_n, x_n) - f(t_n, y_n)\| + \|\delta_n\| \\ &\leq \|x_n - y_n\| + \Delta t L_f \|x_n - y_n\| + \|\delta_n\| \\ &= (1 + L_f \Delta t) \|x_n - y_n\| + \|\delta_n\| \end{aligned}$$

Posons $e_n = \|x_n - y_n\|$ et $\gamma_n = \|\delta_n\|$. On a $e_{n+1} \leq (1 + L_f \Delta t) e_n + \gamma_n$. On utilise une version du lemme de Gronwall discret.

Lemme 10.2.11 (Lemme de Gronwall discret). Soient $(\alpha_n)_{0 \leq n \leq N}$, $(\gamma_n)_{0 \leq n \leq N-1}$ des suites de réels positifs ou nuls, et $(\beta_n)_{0 \leq n \leq N-1}$ une suite de réels positifs ou nuls. Si $\alpha_{n+1} \leq (1 + \beta_n) \alpha_n + \gamma_n$ pour $n = 0, \dots, N-1$, alors :

$$\alpha_n \leq \left(\prod_{j=0}^{n-1} (1 + \beta_j) \right) \alpha_0 + \sum_{j=0}^{n-1} \left(\prod_{k=j+1}^{n-1} (1 + \beta_k) \right) \gamma_j$$

En utilisant $1 + u \leq e^u$, on a $\prod (1 + \beta_j) \leq \exp(\sum \beta_j)$. Ainsi, $\alpha_n \leq \exp\left(\sum_{j=0}^{n-1} \beta_j\right) \left(\alpha_0 + \sum_{j=0}^{n-1} \gamma_j\right)$ (en majorant $\exp(-\sum \beta_k) \leq 1$ dans une version plus précise). Plus précisément, si $\alpha_{n+1} \leq (1 + B) \alpha_n + \Gamma$ pour $B \geq 0$, alors $\alpha_n \leq (1 + B)^n \alpha_0 + \Gamma \sum_{j=0}^{n-1} (1 + B)^j \leq e^{nB} \alpha_0 + \Gamma \frac{e^{nB} - 1}{B}$. Si Γ_j varie: $\alpha_n \leq e^{\sum_{j=0}^{n-1} \beta_j} \left(\alpha_0 + \sum_{j=0}^{n-1} \gamma_j e^{-\sum_{k=0}^j \beta_k} \right)$.

Dans notre cas, $\beta_n = L_f \Delta t$. Donc $\sum_{j=0}^{n-1} \beta_j = n L_f \Delta t$. Comme $n \Delta t \leq N \Delta t = T$:

$$e^{\sum_{j=0}^{n-1} L_f \Delta t} = e^{n L_f \Delta t} \leq e^{L_f T}$$

Aussi, $e^{-\sum_{k=0}^j L_f \Delta t} \leq 1$. Donc, pour tout $n \in \{0, \dots, N\}$:

$$\|x_n - y_n\| \leq e^{L_f T} \left(\|x_0 - y_0\| + \sum_{j=0}^{N-1} \|\delta_j\| \right)$$

Le schéma d'Euler explicite est donc stable avec une constante de stabilité $S = e^{L_f T}$. \square

Proposition 10.2.12 (Condition suffisante de stabilité pour Φ Lipschitzienne). Si la fonction d'incrément $\Phi(t, y, \Delta t)$ est Lipschitzienne par rapport à y , uniformément en t et Δt , c'est-à-dire s'il existe une constante $\Lambda > 0$ telle que pour tous $y_1, y_2 \in \mathbb{R}^d$, pour tout $t \in [t_0, t_0 + T]$ et pour tout $\Delta t \in [0, \Delta t_0]$:

$$\|\Phi(t, y_1, \Delta t) - \Phi(t, y_2, \Delta t)\| \leq \Lambda \|y_1 - y_2\|$$

alors le schéma (S) $x_{n+1} = x_n + \Delta t \Phi(t_n, x_n, \Delta t)$ est stable, avec une constante de stabilité $S = e^{\Lambda T}$.

Preuve. La preuve est analogue à celle pour le schéma d'Euler explicite. On a :

$$\begin{aligned} \|x_{n+1} - y_{n+1}\| &\leq \|x_n - y_n + \Delta t(\Phi(t_n, x_n, \Delta t) - \Phi(t_n, y_n, \Delta t))\| + \|\delta_n\| \\ &\leq \|x_n - y_n\| + \Delta t \|\Phi(t_n, x_n, \Delta t) - \Phi(t_n, y_n, \Delta t)\| + \|\delta_n\| \\ &\leq \|x_n - y_n\| + \Delta t \Lambda \|x_n - y_n\| + \|\delta_n\| \\ &= (1 + \Lambda \Delta t) \|x_n - y_n\| + \|\delta_n\| \end{aligned}$$

L'application du lemme de Gronwall discret donne le résultat avec $S = e^{\Lambda T}$. □

10.2.4 Convergence des schémas à un pas explicite

Le théorème suivant (parfois appelé Théorème de Lax pour les EDO) relie consistance, stabilité et convergence.

Theorem 10.2.13 (Convergence des schémas à un pas explicites). Si la méthode à un pas explicite (S) est stable et consistante d'ordre $q \geq 1$, alors elle est convergente. De plus, si la solution $x(t)$ du problème de Cauchy (P) est de classe C^{q+1} et si la suite $(x_n)_{0 \leq n \leq N}$ est générée par (S) avec $x_0 = x(t_0)$, alors il existe une constante $K > 0$ telle que :

$$\max_{0 \leq n \leq N} \|x(t_n) - x_n\| \leq K(\Delta t)^q$$

Plus précisément, si C_{cons} est la constante de l'erreur de consistance (i.e. $\|e_{n+1}\| \leq C_{cons}(\Delta t)^{q+1}$) et S est la constante de stabilité, on a :

$$\max_{0 \leq n \leq N} \|x(t_n) - x_n\| \leq S \sum_{j=0}^{N-1} \|e_{j+1}\| \approx S \cdot N \cdot C_{cons}(\Delta t)^{q+1} = S \cdot \frac{T}{\Delta t} \cdot C_{cons}(\Delta t)^{q+1} = S \cdot C_{cons} \cdot T \cdot (\Delta t)^q$$

L'erreur globale est donc en $O((\Delta t)^q)$.

Preuve (Idée de la preuve). Soit $x(t)$ la solution exacte. On a $x(t_{n+1}) = x(t_n) + \Delta t \Phi(t_n, x(t_n), \Delta t) + e_{n+1}$, où e_{n+1} est l'erreur locale de troncature. Le schéma numérique est $x_{n+1} = x_n + \Delta t \Phi(t_n, x_n, \Delta t)$. Soit $\varepsilon_n = x(t_n) - x_n$ l'erreur globale au temps t_n . $\varepsilon_{n+1} = x(t_{n+1}) - x_{n+1} = (x(t_n) - x_n) + \Delta t(\Phi(t_n, x(t_n), \Delta t) - \Phi(t_n, x_n, \Delta t)) + e_{n+1}$. En utilisant la condition de Lipschitz pour Φ (qui découle de la stabilité pour de nombreux schémas ou est une hypothèse pour la stabilité plus générale): $\|\varepsilon_{n+1}\| \leq \|\varepsilon_n\| + \Delta t \Lambda \|\varepsilon_n\| + \|e_{n+1}\| = (1 + \Lambda \Delta t) \|\varepsilon_n\| + \|e_{n+1}\|$. Puisque $x_0 = x(t_0)$, $\varepsilon_0 = 0$. L'erreur locale $\|e_{n+1}\| \leq C_{cons}(\Delta t)^{q+1}$. Par le lemme de Gronwall, avec $\alpha_n = \|\varepsilon_n\|$, $\beta_n = \Lambda \Delta t$, $\gamma_n = C_{cons}(\Delta t)^{q+1}$:

$$\|\varepsilon_n\| \leq e^{n\Lambda\Delta t} \left(\|\varepsilon_0\| + \sum_{j=0}^{n-1} C_{cons}(\Delta t)^{q+1} e^{-(j+1)\Lambda\Delta t} \right)$$

En majorant $e^{-(j+1)\Lambda\Delta t} \leq 1$ et $e^{n\Lambda\Delta t} \leq e^{\Lambda T}$:

$$\|\varepsilon_n\| \leq e^{\Lambda T} \sum_{j=0}^{n-1} C_{cons}(\Delta t)^{q+1} = e^{\Lambda T} n C_{cons}(\Delta t)^{q+1}$$

Comme $n\Delta t \leq T$, on a $n \leq T/\Delta t$.

$$\|\varepsilon_n\| \leq e^{\Lambda T} \frac{T}{\Delta t} C_{cons}(\Delta t)^{q+1} = (e^{\Lambda T} T C_{cons})(\Delta t)^q$$

Ceci montre que $\max_{0 \leq n \leq N} \|\varepsilon_n\| = O((\Delta t)^q)$. La constante S du théorème est $e^{\Lambda T}$ et la constante c est C_{cons} . \square

Chapter 11

CM11

Méthodes de Résolution Numérique des Équations

Chapitre 1 : Introduction et Méthodes de Base

La résolution d'équations, notamment non linéaires, est une tâche fondamentale en sciences et ingénierie. Lorsque les solutions analytiques ne sont pas disponibles ou pratiques à obtenir, les méthodes numériques offrent des outils puissants pour approximer ces solutions. Ce chapitre introduit les concepts de base relatifs à la recherche de racines d'équations et présente plusieurs méthodes numériques fondamentales, telles que la dichotomie et la méthode de la fausse position. Nous aborderons également les notions d'existence, d'unicité des solutions, ainsi que la convergence de ces algorithmes.

11.1 Généralités et exemples

11.1.1 Rappels ((EO), $f(x)=0$)

Une équation est une instruction mathématique qui stipule que deux expressions sont égales. La résolution d'une équation consiste à trouver les valeurs des inconnues pour lesquelles cette égalité est vérifiée. Dans ce chapitre, nous nous intéressons aux équations de la forme :

$$f(x) = 0$$

où f est une fonction définie sur un domaine $D \subseteq \mathbb{R}$ à valeurs dans \mathbb{R} . On cherche $x \in D$ tel que $f(x) = 0$. Une telle équation est parfois notée (EO).

Definition 11.1.1 (Équation linéaire). Si la fonction f est affine, c'est-à-dire de la forme $f(x) = ax + b$ avec $a, b \in \mathbb{R}$, l'équation $f(x) = 0$ est appelée **équation linéaire**. Si $a \neq 0$, la solution unique est $x = -b/a$. Si $a = 0$ et $b \neq 0$, il n'y a pas de solution. Si $a = 0$ et $b = 0$, tout $x \in \mathbb{R}$ est solution. Les équations qui ne sont pas linéaires sont dites **non linéaires**.

Definition 11.1.2 (Solution ou Racine). Tout $x^* \in D$ tel que $f(x^*) = 0$ est appelé une **solution** de l'équation (EO). Une solution est également appelée une **racine** ou un **zéro** de la fonction f . L'ensemble de définition de f est noté D_f .

11.1.2 Définitions

Considérons une fonction f de classe $\mathcal{C}^p(D, \mathbb{R})$ pour $p \geq 1$, où D est un intervalle de \mathbb{R} . Soit $x^* \in D$ une racine de f , i.e. $f(x^*) = 0$.

Definition 11.1.3 (Racine simple). La racine x^* est dite **simple** si $f'(x^*) \neq 0$.

Definition 11.1.4 (Racine multiple). La racine x^* est dite de **multiplicité** p (où $p \geq 1$ est un entier) si

$$f(x^*) = f'(x^*) = \dots = f^{(p-1)}(x^*) = 0 \quad \text{et} \quad f^{(p)}(x^*) \neq 0.$$

Une racine simple est donc une racine de multiplicité $p = 1$.

11.1.3 Exemples

Certains problèmes mènent à la résolution d'équations non linéaires. Par exemple, dans la résolution numérique d'équations différentielles ordinaires (EDO) de la forme $x'(t) = f(t, x(t))$, les schémas implicites nécessitent de résoudre une équation à chaque pas de temps.

Example 11.1.5 (Euler implicite). Le schéma d'Euler implicite pour l'EDO $x'(t) = f(t, x(t))$ est donné par :

$$x_{n+1} = x_n + hf(t_{n+1}, x_{n+1})$$

où x_n est une approximation de $x(t_n)$, h est le pas de temps, et $t_{n+1} = t_n + h$. Pour déterminer x_{n+1} à l'itération n , il faut résoudre l'équation $g_n(z) = 0$ pour z , où

$$g_n(z) = z - x_n - hf(t_{n+1}, z).$$

Cette équation est généralement non linéaire en $z \equiv x_{n+1}$.

D'autres schémas numériques sont des exemples de suites récurrentes :

Example 11.1.6 (Schéma de Crank-Nicolson type (CN)). Soit $x_0 = x_0^*$ (valeur initiale). La suite est définie par :

$$x_{n+1} = x_n + \frac{h}{2}[f(t_n, x_n) + f(t_{n+1}, x_{n+1}^*)] \quad \text{pour } n = 0, \dots, N-1.$$

La valeur x_{n+1}^* doit être définie. Si $x_{n+1}^* = x_{n+1}$, le schéma est implicite (Crank-Nicolson standard) et nécessite la résolution d'une équation pour x_{n+1} . Si x_{n+1}^* est une prédiction explicite de x_{n+1} , le schéma peut devenir explicite.

Example 11.1.7 (Schéma de Heun). Ce schéma est un exemple de méthode de prédicteur-correcteur :

$$\begin{aligned} x_0 &= x_0^* \quad (\text{valeur initiale}) \\ x_{n+1}^* &= x_n + hf(t_n, x_n) \quad (\text{prédicteur}) \\ x_{n+1} &= x_n + \frac{h}{2}[f(t_n, x_n) + f(t_{n+1}, x_{n+1}^*)] \quad (\text{correcteur}) \end{aligned}$$

Ce schéma, tel qu'écrit, est explicite et ne requiert pas la résolution d'une équation pour x_{n+1} à chaque étape, car x_{n+1}^* est calculé en premier.

11.2 Partition Correcte du Problème (EO)

Lorsqu'on aborde la résolution de l'équation $f(x) = 0$, les premières questions qui se posent sont :

- L'équation (EO) admet-elle une solution ? (Existence)
- Si oui, cette solution est-elle unique ? (Unicité)

De plus, la dépendance continue de la solution par rapport aux données du problème (stabilité) est une propriété souhaitable. Pour simplifier, nous nous plaçons dans le cadre scalaire où $f : \mathbb{R} \rightarrow \mathbb{R}$.

11.2.1 Existence et unicité de la solution

Proposition 11.2.1 (Cas $f(x) = 0$). Soit f une fonction continue sur un intervalle fermé $I = [a, b]$.

- Si $f(a)f(b) < 0$ (c'est-à-dire si $f(a)$ et $f(b)$ sont de signes opposés), alors il existe au moins une racine $x^* \in]a, b[$ telle que $f(x^*) = 0$. (Ceci découle du Théorème des Valeurs Intermédiaires).
- Si de plus f est strictement monotone sur I , alors cette racine x^* est unique dans I .

Preuve (esquisse) : L'existence vient du Théorème des Valeurs Intermédiaires. Pour l'unicité, si f est strictement monotone, elle est injective sur I . Donc, si $x_1 \neq x_2$ dans I , alors $f(x_1) \neq f(x_2)$. Ainsi, $f(x) = 0$ ne peut avoir plus d'une solution.

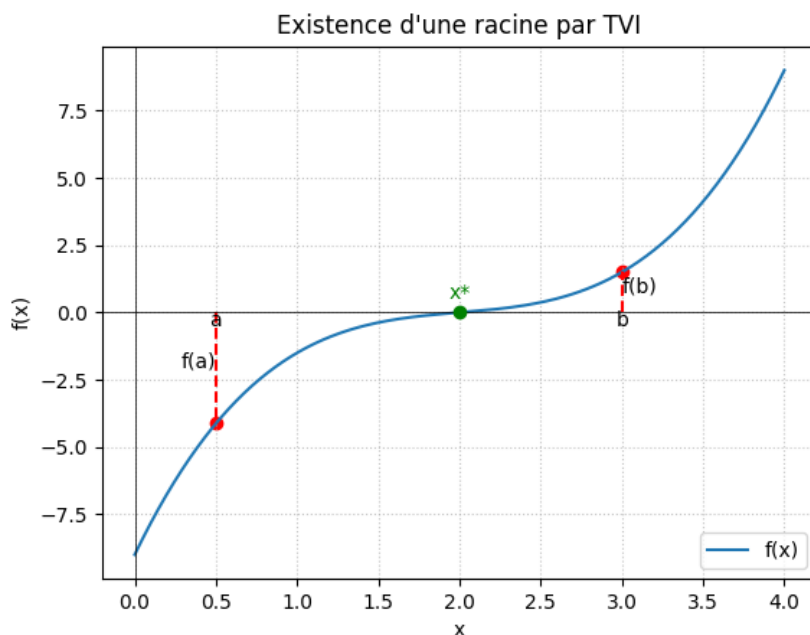


Figure 11.1: Illustration du Théorème des Valeurs Intermédiaires : si $f(a)f(b) < 0$ et f est continue, il existe $x^* \in]a, b[$ tel que $f(x^*) = 0$.

Un autre type courant d'équation est l'équation de point fixe $x = g(x)$.

Proposition 11.2.2 (Cas $x = g(x)$). Soit g une fonction continue et I un intervalle fermé de \mathbb{R} tel que $g(I) \subseteq I$ (c'est-à-dire que g envoie I dans lui-même).

- Alors il existe au moins un point fixe $x^* \in I$ tel que $g(x^*) = x^*$.
- Si de plus g est contractante sur I , c'est-à-dire s'il existe $K \in [0, 1)$ tel que pour tous $x, y \in I$, $|g(x) - g(y)| \leq K|x - y|$, alors le point fixe x^* est unique dans I .
- Si g est dérivable sur I et s'il existe $K \in [0, 1)$ tel que $|g'(x)| \leq K$ pour tout $x \in I$, alors g est contractante sur I (par le théorème des accroissements finis).

11.2.2 Propriétés utiles

Remark 11.2.3 (Multiplicité des racines). Lorsque la racine cherchée x^* est de multiplicité $p > 1$ (racine multiple), des difficultés numériques peuvent apparaître. Par exemple, si $f'(x^*) = 0$, la pente de la fonction est nulle à la racine, ce qui peut ralentir la convergence de certaines méthodes ou les rendre instables. Numériquement, il sera souvent plus difficile de localiser précisément une racine multiple qu'une racine simple.

11.3 Construction de schémas pour (EO)

11.3.1 Méthode de dichotomie (cas $f(x) = 0$)

La méthode de dichotomie (ou méthode de bisection) est l'une des méthodes les plus simples et robustes pour trouver une racine d'une fonction continue.

Principe : On part d'un intervalle initial $I_0 = [a_0, b_0]$ tel que $f(a_0)f(b_0) < 0$. On est donc assuré par le TVI qu'il existe au moins une racine dans $]a_0, b_0[$. À chaque étape n , on considère l'intervalle $[a_n, b_n]$:

1. On calcule le milieu de l'intervalle : $c_n = \frac{a_n + b_n}{2}$.
2. On évalue $f(c_n)$.
3. Si $f(c_n) = 0$ (ou est numériquement proche de zéro), alors c_n est la racine (ou une bonne approximation).
4. Sinon :
 - Si $f(a_n)f(c_n) < 0$, la racine se trouve dans $[a_n, c_n]$. On pose alors $[a_{n+1}, b_{n+1}] = [a_n, c_n]$.
 - Si $f(c_n)f(b_n) < 0$, la racine se trouve dans $[c_n, b_n]$. On pose alors $[a_{n+1}, b_{n+1}] = [c_n, b_n]$. (Note: si $f(a_n)f(c_n) > 0$ et $f(c_n) \neq 0$, alors $f(c_n)f(b_n)$ doit être < 0 car $f(a_n)$ et $f(b_n)$ sont de signes opposés).

On répète ce processus, divisant la taille de l'intervalle de recherche par deux à chaque étape.

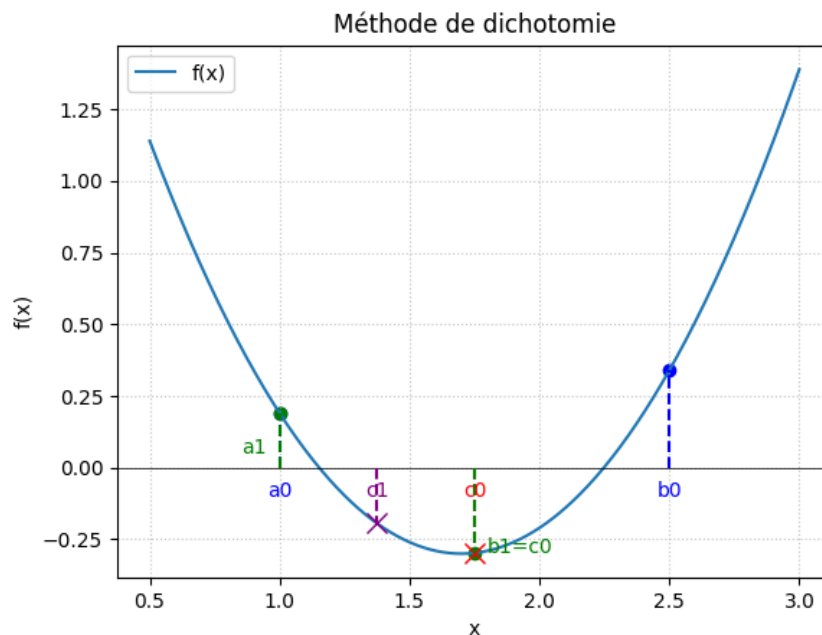


Figure 11.2: Illustration de la méthode de dichotomie. L'intervalle $[a, b]$ est divisé par deux à chaque étape.

Algorithme de dichotomie :

1. Initialisation :

- Choisir a, b tels que $f(a)f(b) < 0$.
- Choisir une tolérance $\varepsilon > 0$ pour l'erreur.
- Choisir un nombre maximal d'itérations k_{\max} .
- Initialiser le compteur d'itérations $k = 0$.

2. Itérations : Tant que $(b - a) > \varepsilon$ (ou $(b - a)/2 > \varepsilon$) et $k < k_{\max}$:

- (a) Calculer $c = (a + b)/2$.
- (b) Si $f(c) = 0$ (ou $|f(c)| < \varepsilon_{\text{res}}$, une tolérance sur le résidu), alors $x^* = c$, arrêter.
- (c) Si $f(a)f(c) < 0$, alors $b = c$.
- (d) Sinon (si $f(c)f(b) < 0$), alors $a = c$.
- (e) $k = k + 1$.

3. Résultat : La racine est approximée par $c = (a + b)/2$. L'erreur absolue $|x^* - c|$ est inférieure à $(b - a)/2$.

Les critères d'arrêt peuvent varier, par exemple : la largeur de l'intervalle $(b_N - a_N) \leq \varepsilon$, la valeur de la fonction $|f(c_N)| \leq \varepsilon_f$, ou un nombre maximal d'itérations atteint.

Propriétés :

- **Convergence** : La méthode de dichotomie converge toujours si f est continue et $f(a_0)f(b_0) < 0$. L'erreur à l'étape n (en prenant c_n comme approximation de x^*) est bornée :

$$|x^* - c_n| \leq \frac{b_n - a_n}{2} = \frac{b_0 - a_0}{2^{n+1}}$$

Comme $(b_0 - a_0)/2^{n+1} \rightarrow 0$ quand $n \rightarrow \infty$, la suite c_n converge vers x^* . La convergence est **linéaire** (ou d'ordre 1), avec un taux de convergence $\mu = 1/2$. C'est-à-dire $\lim_{n \rightarrow \infty} \frac{|x^* - c_{n+1}|}{|x^* - c_n|} = \frac{1}{2}$ si c_n est pris comme le milieu de l'intervalle qui devient le point final de l'intervalle suivant.

- **Nombre d'itérations** : Pour atteindre une précision ε (i.e., $b_N - a_N \leq \varepsilon$), le nombre d'itérations N nécessaire est $N \approx \log_2 \left(\frac{b_0 - a_0}{\varepsilon} \right)$.
- **Robustesse** : La méthode est très robuste car elle est garantie de converger.
- **Inconvénients** : La convergence peut être lente comparée à d'autres méthodes, surtout si la précision requise est élevée. Elle nécessite de nombreux calculs de $f(x)$ (un par itération, plus les deux initiaux).

11.3.2 Méthode de la fausse position (Regula Falsi)

La méthode de la fausse position (ou *Regula Falsi*) est une autre méthode de bracketing pour trouver une racine. Elle est similaire à la dichotomie, mais au lieu de diviser l'intervalle en deux, elle utilise une interpolation linéaire.

Principe : Comme pour la dichotomie, on part d'un intervalle $[a_n, b_n]$ tel que $f(a_n)f(b_n) < 0$. Au lieu de prendre le milieu $c_n = (a_n + b_n)/2$, on approxime f par la droite passant par les points $(a_n, f(a_n))$ et $(b_n, f(b_n))$. Le point c_n est alors l'intersection de cette droite avec l'axe des abscisses. L'équation de la droite est $y - f(a_n) = \frac{f(b_n) - f(a_n)}{b_n - a_n}(x - a_n)$. En posant $y = 0$ et $x = c_n$, on obtient :

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

Ensuite, comme pour la dichotomie, on compare les signes de $f(a_n)f(c_n)$ et $f(c_n)f(b_n)$ pour déterminer le nouvel intervalle $[a_{n+1}, b_{n+1}]$ qui contient la racine.

Algorithme de la fausse position : L'algorithme est similaire à celui de la dichotomie, la seule différence étant le calcul de c_n :

1. **Initialisation** : Idem dichotomie.

2. **Itérations** : Tant que $(b - a) > \varepsilon$ (ou une autre condition d'arrêt) et $k < k_{\max}$:

- (a) Calculer $c = \frac{af(b)-bf(a)}{f(b)-f(a)}$.
- (b) Si $f(c) = 0$ (ou $|f(c)| < \varepsilon_{\text{res}}$), alors $x^* = c$, arrêter.
- (c) Si $f(a)f(c) < 0$, alors $b = c$.
- (d) Sinon (si $f(c)f(b) < 0$), alors $a = c$.
- (e) $k = k + 1$.

3. **Résultat** : La racine est approximée par c .

Le coût par itération est d'une évaluation de f .

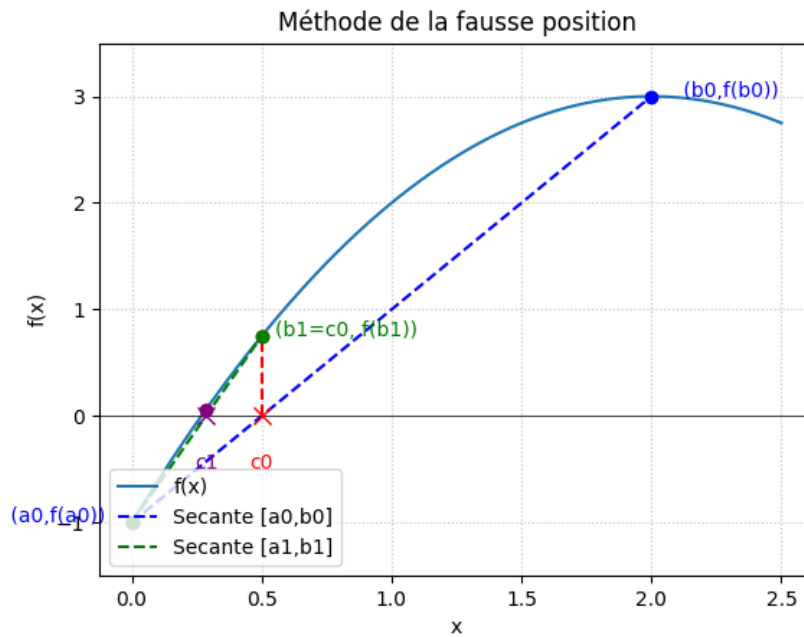


Figure 11.3: Illustration de la méthode de la fausse position. Le point c_n est l'intersection de la sécante avec l'axe des x . Un des points d'extrémité de l'intervalle (a_n ici) peut rester fixe pendant plusieurs itérations si la fonction a une convexité/concavité marquée.

Convergence de la fausse position :

- En général, la fausse position converge plus vite que la dichotomie. Cependant, sa convergence reste linéaire.
- Une particularité est que si la fonction f est convexe ou concave sur $[a, b]$ (c'est-à-dire si f'' ne change pas de signe sur $[a, b]$), l'un des points d'extrémité de l'intervalle (a_n ou b_n) restera fixe après quelques itérations (ou dès la première). L'autre extrémité convergera vers la racine x^* .

Proposition 11.3.1. Si $f \in \mathcal{C}^2([a, b])$, $f(a)f(b) < 0$, et si f'' n'a aucune racine dans $]a, b[$ (i.e., f est strictement convexe ou strictement concave sur $[a, b]$), alors une des suites (a_n) ou (b_n) demeure constante à partir d'un certain rang. La suite (c_n) générée par la méthode de la fausse position converge linéairement vers la racine x^* de f .

- Si, par exemple, b_n est constant (égal à b) à partir d'un certain rang, alors la suite a_n converge vers x^* , et l'erreur $e_n = x^* - a_n$ satisfait :

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

(car $f(x^*) = 0$). Un résultat similaire s'applique si a_n est constant et $b_n \rightarrow x^*$.

Bien que souvent plus rapide que la dichotomie en termes de nombre d'itérations, la convergence peut être lente si le point fixe est très éloigné de la racine par rapport à l'autre point. Des variantes existent pour améliorer ce comportement (ex: méthode de l'Illinois).

11.4 Algorithmes : Méthodes Itératives

Les méthodes de dichotomie et de fausse position sont des exemples de méthodes itératives. Une méthode itérative est un procédé mathématique qui génère une suite de solutions approchées $(x_n)_{n \in \mathbb{N}}$ dans le but de converger vers la solution exacte x^* .

11.4.1 Convergence

Pour une méthode itérative générant une suite (x_n) qui converge vers une racine x^* , on s'intéresse à la vitesse de convergence.

Definition 11.4.1 (Ordre de convergence). Soit $(x_n)_{n \in \mathbb{N}}$ une suite qui converge vers x^* . On dit que la convergence est d'ordre $p \geq 1$ s'il existe une constante $\mu > 0$ (appelée taux de convergence asymptotique) telle que :

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

- Si $p = 1$ et $0 < \mu < 1$, la convergence est dite **linéaire**.
- Si $p = 1$ et $\mu = 0$, la convergence est dite **superlinéaire**.
- Si $p = 2$, la convergence est dite **quadratique**.
- Si $p = 3$, la convergence est dite **cubique**.

Plus l'ordre p est élevé, plus la convergence est rapide (en général). Si $p = 1$, un μ petit indique une convergence plus rapide.

Example 11.4.2 (Convergence de la dichotomie). Comme mentionné précédemment, la méthode de dichotomie a une convergence linéaire ($p = 1$) avec un taux $\mu = 1/2$. En notant $e_n = |x^* - c_n|$, on a $e_{n+1} \approx \frac{1}{2}e_n$.

Example 11.4.3 (Convergence de la fausse position). La méthode de la fausse position a également une convergence linéaire ($p = 1$). Le taux de convergence μ dépend de la fonction f et de l'intervalle initial (voir la formule donnée précédemment). Dans de nombreux cas, μ est plus petit que $1/2$, ce qui rend la fausse position plus rapide que la dichotomie, mais elle peut être très proche de 1 dans des cas défavorables (si le point fixe est "mauvais").

Chapter 12

CM12

12.1 Méthode d'itération

12.1.1 Principe

La méthode d'itération, ou méthode du point fixe, est utilisée pour résoudre des équations de la forme $f(x) = 0$. Elle est adaptée aux problèmes où l'on peut transformer l'équation $f(x) = 0$ en une forme équivalente $x = g(x)$. On cherche alors une valeur x^* telle que $x^* = g(x^*)$. Une telle valeur x^* est appelée un point fixe de la fonction g . La méthode consiste à construire une suite $(x_n)_{n \in \mathbb{N}}$ définie par la relation de récurrence :

$$(S) \quad \begin{cases} x_0 \text{ donné} \\ x_{n+1} = g(x_n), \quad \forall n \geq 0 \end{cases} \quad (12.1)$$

Si cette suite converge vers une limite x^* , et si g est continue, alors x^* est un point fixe de g . En effet, $x^* = \lim_{n \rightarrow \infty} x_{n+1} = \lim_{n \rightarrow \infty} g(x_n) = g(\lim_{n \rightarrow \infty} x_n) = g(x^*)$.

12.1.2 Représentation graphique

Graphiquement, les points fixes de g sont les abscisses des points d'intersection de la courbe $y = g(x)$ avec la droite $y = x$. Pour construire la suite (x_n) , on part de x_0 .

1. On trouve $g(x_0)$ sur la courbe $y = g(x)$. Ce point a pour coordonnées $(x_0, g(x_0))$. Comme $x_1 = g(x_0)$, ce point est (x_0, x_1) .
2. Pour reporter x_1 sur l'axe des abscisses, on se déplace horizontalement jusqu'à la droite $y = x$. Le point atteint est (x_1, x_1) .
3. On trouve $g(x_1)$ sur la courbe $y = g(x)$ en se déplaçant verticalement. Ce point est $(x_1, g(x_1))$, c'est-à-dire (x_1, x_2) .
4. On répète le processus.

Selon la configuration des courbes, la construction peut donner un motif en “escalier” ou en “spirale” (ou “escargot”).

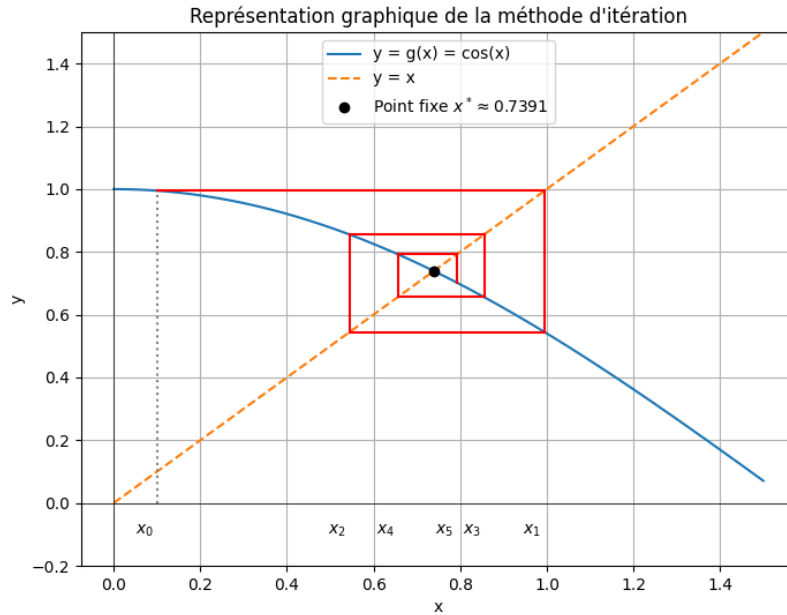


Figure 12.1: Illustration de la méthode du point fixe pour $g(x) = \cos(x)$ et $x_0 = 0.1$. La suite (x_n) converge vers le point fixe $x^* \approx 0.739$.

12.1.3 Algorithme

L'algorithme de la méthode du point fixe est le suivant :

1. Initialisation :

- Choisir un point de départ x_0 .
- Se donner une tolérance $\varepsilon > 0$ (pour le critère d'arrêt).
- Définir un nombre maximal d'itérations N_{max} .
- Poser $n = 0$.

2. Itérations :

- Tant que $n < N_{max}$ (et critère d'arrêt non satisfait) :
 - (a) Calculer $x_{n+1} = g(x_n)$.
 - (b) Vérifier le critère d'arrêt. Par exemple, si $|x_{n+1} - x_n| < \varepsilon$, alors arrêter. D'autres critères peuvent être $|x_{n+1} - x_n|/|x_{n+1}| < \varepsilon$ (si $x_{n+1} \neq 0$) ou $|f(x_{n+1})| < \varepsilon$. La note manuscrite mentionne $E_{n+1} = |x_{n+1} - x_n|$.
 - (c) $n = n + 1$.
 - (d) Mettre à jour $x_n \leftarrow x_{n+1}$ pour la prochaine itération (ou $x_{old} \leftarrow x_n$, $x_n \leftarrow x_{n+1}$).
- Fin Tant que.

3. **Arrêt :** La valeur x_n (ou x_{n+1}) obtenue est une approximation du point fixe x^* . Si le nombre maximal d'itérations N_{max} est atteint sans que le critère de tolérance soit satisfait, la méthode peut avoir échoué à converger ou converger trop lentement.

La note manuscrite mentionne : "Tant que $E_n > E$ et N_{max} pas grand (atteint)".

12.1.4 Convergence

Remark 12.1.1. Il y a une similitude entre les racines de la fonction $h(x) = g(x) - x$ et les points fixes de $g(x)$. Un point x^* est un point fixe de g si et seulement si x^* est une racine de $h(x) = g(x) - x = 0$.

Proposition 12.1.2. Soit I un intervalle fermé de \mathbb{R} . Soit $g : I \rightarrow \mathbb{R}$ une fonction telle que :

1. $g(I) \subset I$ (c'est-à-dire, pour tout $x \in I$, $g(x) \in I$).
2. g est contractante sur I , c'est-à-dire qu'il existe une constante $K \in [0, 1)$ telle que pour tous $x, y \in I$, $|g(x) - g(y)| \leq K|x - y|$.

Alors :

1. g admet un unique point fixe x^* dans I .
2. Pour tout choix initial $x_0 \in I$, la suite (x_n) définie par $x_{n+1} = g(x_n)$ converge vers x^* .
3. On a les estimations d'erreur suivantes :

- $|x_n - x^*| \leq K^n |x_0 - x^*|$
- $|x_n - x^*| \leq \frac{K^n}{1-K} |x_1 - x_0|$

Si g est dérivable sur I , la condition de contraction (2) est satisfaite si $\sup_{x \in I} |g'(x)| \leq K < 1$. Si $|g'(x^*)| > 1$, la méthode diverge (sauf si $x_0 = x^*$).

Preuve. (Suivant les notes manuscrites) **Existence :** Soit $I = [a, b]$. Posons $h(x) = g(x) - x$. Comme $g(I) \subset I$, on a $g(a) \in [a, b]$ et $g(b) \in [a, b]$. Donc $g(a) \geq a \implies h(a) = g(a) - a \geq 0$. Et $g(b) \leq b \implies h(b) = g(b) - b \leq 0$. Si $h(a) = 0$, alors a est un point fixe. Si $h(b) = 0$, alors b est un point fixe. Sinon, si $h(a) > 0$ et $h(b) < 0$, et g (donc h) est continue (car dérivable, ou contractante implique continue), d'après le Théorème des Valeurs Intermédiaires, il existe $x^* \in (a, b)$ tel que $h(x^*) = 0$, c'est-à-dire $g(x^*) = x^*$.

Unicité : Supposons qu'il existe deux points fixes distincts x^* et x^{**} dans I . Alors $g(x^*) = x^*$ et $g(x^{**}) = x^{**}$. On a $|x^{**} - x^*| = |g(x^{**}) - g(x^*)|$. Si g est contractante avec une constante $K < 1$, alors $|g(x^{**}) - g(x^*)| \leq K|x^{**} - x^*|$. Donc $|x^{**} - x^*| \leq K|x^{**} - x^*|$. Comme $x^* \neq x^{**}$, $|x^{**} - x^*| > 0$. On peut diviser par $|x^{**} - x^*|$ pour obtenir $1 \leq K$. Ceci contredit $K < 1$. Donc, l'hypothèse qu'il existe deux points fixes distincts est fautive. Le point fixe est unique.

Convergence : Soit $x_0 \in I$. La suite (x_n) est définie par $x_{n+1} = g(x_n)$. On a $x^* = g(x^*)$. Alors $|x_{n+1} - x^*| = |g(x_n) - g(x^*)|$. En utilisant la propriété de contraction (ou le théorème des accroissements finis si g est dérivable, $|g(x_n) - g(x^*)| \leq |g'(c_n)||x_n - x^*|$ pour un c_n entre x_n et x^* , avec $|g'(c_n)| \leq K$), on a : $|x_{n+1} - x^*| \leq K|x_n - x^*|$. Par récurrence, on obtient : $|x_n - x^*| \leq K^n|x_0 - x^*|$. Comme $0 \leq K < 1$, $K^n \rightarrow 0$ quand $n \rightarrow \infty$. Donc, $\lim_{n \rightarrow \infty} |x_n - x^*| = 0$, ce qui signifie que la suite (x_n) converge vers x^* . La note mentionne $K_1 = \lim_{n \rightarrow \infty} \frac{x_{n+1} - x^*}{x_n - x^*} = g'(x^*)$ et $|K_1| \leq K < 1$. Ceci est lié à l'ordre de convergence. \square

Ordre de convergence

L'ordre de convergence d'une suite (x_n) vers x^* est un nombre $p \geq 1$ tel que :

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

où C est la constante asymptotique d'erreur. Si $p = 1$, la convergence est linéaire. Si $p = 2$, elle est quadratique.

Supposons que g est suffisamment dérivable et utilisons un développement de Taylor de $g(x_n)$ autour de x^* : $x_{n+1} = g(x_n) = g(x^*) + g'(x^*)(x_n - x^*) + \frac{g''(x^*)}{2!}(x_n - x^*)^2 + \dots + \frac{g^{(k)}(x^*)}{k!}(x_n - x^*)^k + O((x_n - x^*)^{k+1})$.
Puisque $x^* = g(x^*)$, on a : $x_{n+1} - x^* = g'(x^*)(x_n - x^*) + \frac{g''(x^*)}{2!}(x_n - x^*)^2 + \dots$.

Si $g'(x^*) \neq 0$, alors :

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

Donc, $\lim_{n \rightarrow \infty} \frac{|x_{n+1} - x^*|}{|x_n - x^*|} = |g'(x^*)|$. La convergence est linéaire (ordre $p = 1$) avec $C = |g'(x^*)|$. Pour que la méthode converge, il faut que $|g'(x^*)| < 1$.

Si $g'(x^*) = 0$, et si g est p -fois dérivable avec $g^{(k)}(x^*) = 0$ for $1 \leq k < p$ et $g^{(p)}(x^*) \neq 0$ (où $p \geq 2$ est le plus petit entier tel que $g^{(p)}(x^*) \neq 0$), alors le développement de Taylor devient : $x_{n+1} - x^* = \frac{g^{(p)}(x^*)}{p!}(x_n - x^*)^p + O((x_n - x^*)^{p+1})$. Dans ce cas :

$$\lim_{n \rightarrow \infty} \frac{|x_{n+1} - x^*|}{|x_n - x^*|^p} = \left| \frac{g^{(p)}(x^*)}{p!} \right|$$

La convergence est d'ordre p , et la constante asymptotique d'erreur est $C = \left| \frac{g^{(p)}(x^*)}{p!} \right|$.

Exemple 12.1.3. Soit $g(x) = e^x - 1 - x - \frac{x^2}{2}$. On cherche un point fixe x^* de g . On observe que $x^* = 0$ est un point fixe, car $g(0) = e^0 - 1 - 0 - 0 = 1 - 1 = 0$. Calculons les dérivées de g en $x^* = 0$: $g'(x) = e^x - 1 - x \implies g'(0) = e^0 - 1 - 0 = 0$. $g''(x) = e^x - 1 \implies g''(0) = e^0 - 1 = 0$. $g'''(x) = e^x \implies g'''(0) = e^0 = 1$. Puisque $g'(0) = 0$, $g''(0) = 0$, et $g'''(0) = 1 \neq 0$, l'ordre de convergence est $p = 3$. La constante asymptotique d'erreur est $C = \left| \frac{g'''(0)}{3!} \right| = \left| \frac{1}{6} \right| = \frac{1}{6}$.

Analyse de la méthode de la fausse position (Regula Falsi).

Solution. Indication : c'est une méthode de point fixe. □

12.2 Méthode de Newton

12.2.1 Principe

La méthode de Newton est une méthode itérative pour trouver une approximation d'une racine x^* d'une fonction f , c'est-à-dire $f(x^*) = 0$. Elle est applicable si f est dérivable.

12.2.2 Comment construire x_{n+1} à partir de x_n ?

On part d'une approximation x_n de la racine x^* . On remplace la fonction $f(x)$ par son polynôme de Taylor d'ordre 1 (sa tangente) au voisinage de x_n :

$$f(x) \approx P_1(x) = f(x_n) + f'(x_n)(x - x_n)$$

On cherche x_{n+1} tel que $P_1(x_{n+1}) = 0$.

$$f(x_n) + f'(x_n)(x_{n+1} - x_n) = 0$$

En supposant $f'(x_n) \neq 0$, on peut résoudre pour x_{n+1} :

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

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

Ceci définit la relation de récurrence de la méthode de Newton. On peut voir cela comme une méthode de point fixe :

$$x_{n+1} = g(x_n)$$

avec $g(x) = x - \frac{f(x)}{f'(x)}$

12.2.3 Interprétation géométrique

Graphiquement, x_{n+1} est l'abscisse du point où la tangente à la courbe $y = f(x)$ au point $(x_n, f(x_n))$ coupe l'axe des abscisses ($y = 0$).

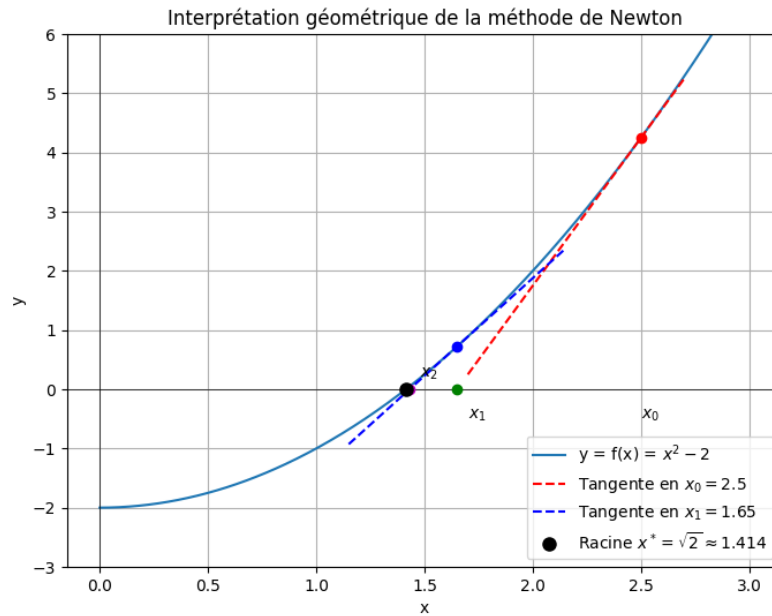


Figure 12.2: Illustration de la méthode de Newton pour $f(x) = x^2 - 2$ avec $x_0 = 2.5$.

12.2.4 Algorithme

L'algorithme de Newton peut être vu comme un cas particulier de l'algorithme du point fixe, avec $g(x) = x - \frac{f(x)}{f'(x)}$.

Listing 12.1: Pseudo-code pour la méthode de Newton

```
def newton_method(f, df, x0, epsilon, N_max):
    # f: la fonction dont on cherche la racine
    # df: la dérivée de f
    # x0: première approximation
    # epsilon: tolérance
    # N_max: nombre maximal d'itérations

    xn = x0
    for n in range(N_max):
        fxn = f(xn)
        if abs(fxn) < epsilon: # Critère d'arrêt sur |f(x_n)|
            print(f"Solution trouvée après {n} itérations.")
            return xn

        dfxn = df(xn)
        if dfxn == 0:
            print("Dérivée nulle. La méthode de Newton échoue.")
            return None # Ou une gestion d'erreur appropriée
```

```

xn_plus_1 = xn - fxn / dfxn

if abs(xn_plus_1 - xn) < epsilon: # Critère d'arrêt sur |x_{n+1}-x_n|
    print(f"Solution trouvée après {n+1} itérations (convergence sur x).")
    return xn_plus_1

xn = xn_plus_1

print("Nombre maximal d'itérations atteint. Pas de convergence.")
return xn # Retourne la dernière approximation

```

La note manuscrite suggère:

```

Algorithme de Newton(f, f', x0, epsilon, N_max):
    g = lambda x: x - f(x)/f'(x)
    return PointFixe(g, x0, epsilon, N_max)
// en utilisant l'algorithme de point fixe défini précédemment

```

12.2.5 Convergence

Pour analyser la convergence de la méthode de Newton, on étudie la fonction d'itération $g(x) = x - \frac{f(x)}{f'(x)}$. On calcule sa dérivée $g'(x)$:

$$\begin{aligned}
 g'(x) &= \frac{d}{dx} \left(x - \frac{f(x)}{f'(x)} \right) \\
 &= 1 - \frac{f'(x)f'(x) - f(x)f''(x)}{(f'(x))^2} \\
 &= \frac{(f'(x))^2 - (f'(x))^2 + f(x)f''(x)}{(f'(x))^2} \\
 &= \frac{f(x)f''(x)}{(f'(x))^2}
 \end{aligned}$$

Si x^* est une racine simple de f (c'est-à-dire $f(x^*) = 0$ et $f'(x^*) \neq 0$), alors :

$$g'(x^*) = \frac{f(x^*)f''(x^*)}{(f'(x^*))^2} = \frac{0 \cdot f''(x^*)}{(f'(x^*))^2} = 0$$

Puisque $g'(x^*) = 0$, d'après la théorie de la méthode du point fixe, si $g''(x^*)$ existe et est non nulle, la convergence est au moins d'ordre 2 (quadratique), à condition que x_0 soit suffisamment proche de x^* . Calculons $g''(x^*)$: Si $g'(x) = \frac{f(x)f''(x)}{(f'(x))^2}$, alors $g''(x) = \frac{(f'(x)f''(x) + f(x)f'''(x))(f'(x))^2 - f(x)f''(x) \cdot 2f'(x)f''(x)}{(f'(x))^4}$. En $x = x^*$, $f(x^*) = 0$, donc: $g''(x^*) = \frac{(f'(x^*)f''(x^*))(f'(x^*))^2 - 0}{(f'(x^*))^4} = \frac{f''(x^*)}{f'(x^*)}$. Si $f''(x^*) \neq 0$ (et $f'(x^*) \neq 0$), alors $g''(x^*) \neq 0$. La convergence de la méthode de Newton est quadratique pour une racine simple, avec $\lim_{n \rightarrow \infty} \frac{|x_{n+1} - x^*|}{|x_n - x^*|^2} = \left| \frac{g''(x^*)}{2} \right| = \left| \frac{f''(x^*)}{2f'(x^*)} \right|$.

Remark 12.2.1 (Cas des racines multiples). Si x^* est une racine de multiplicité $m > 1$, c'est-à-dire $f(x^*) = f'(x^*) = \dots = f^{(m-1)}(x^*) = 0$ et $f^{(m)}(x^*) \neq 0$. Dans ce cas, la convergence de la méthode de Newton redevient linéaire. On peut écrire $f(x) = (x - x^*)^m h(x)$ avec $h(x^*) \neq 0$. Alors $f'(x) = m(x - x^*)^{m-1}h(x) + (x - x^*)^m h'(x)$. La fonction d'itération $g(x) = x - \frac{f(x)}{f'(x)}$ devient :

$$\begin{aligned}
 g(x) &= x - \frac{(x - x^*)^m h(x)}{m(x - x^*)^{m-1}h(x) + (x - x^*)^m h'(x)} \\
 &= x - \frac{(x - x^*)h(x)}{mh(x) + (x - x^*)h'(x)}
 \end{aligned}$$

Pour évaluer $g'(x^*)$, on peut calculer la limite de $g'(x)$ quand $x \rightarrow x^*$, ou utiliser une forme plus

complexe de $g'(x)$. La note indique que $g'(x^*) = 1 - \frac{1}{m}$. En effet, $g'(x) = \frac{d}{dx} \left(x - \frac{(x-x^*)h(x)}{mh(x)+(x-x^*)h'(x)} \right)$. En $x = x^*$, après un calcul (assez long, ou en utilisant la limite $\lim_{x \rightarrow x^*} \frac{f(x)f''(x)}{(f'(x))^2}$ avec les formes de f, f', f'' pour une racine multiple), on trouve $g'(x^*) = 1 - \frac{1}{m}$. Puisque $m > 1$, $0 \leq g'(x^*) < 1$. La convergence est donc linéaire avec une constante asymptotique $K_1 = 1 - \frac{1}{m}$.

Exemple 12.2.2. Soit $f(x) = (x-1)^3$. Ici $x^* = 1$ est une racine de multiplicité $m = 3$. $f(1) = 0$, $f'(x) = 3(x-1)^2 \implies f'(1) = 0$, $f''(x) = 6(x-1) \implies f''(1) = 0$, $f'''(x) = 6 \implies f'''(1) = 6 \neq 0$. La méthode de Newton standard donne : $x_{n+1} = x_n - \frac{(x_n-1)^3}{3(x_n-1)^2} = x_n - \frac{x_n-1}{3} = \frac{2}{3}x_n + \frac{1}{3}$. C'est une suite arithmético-géométrique. Le point fixe est $x^* = \frac{1/3}{1-2/3} = 1$. $x_{n+1} - 1 = \frac{2}{3}x_n + \frac{1}{3} - 1 = \frac{2}{3}x_n - \frac{2}{3} = \frac{2}{3}(x_n - 1)$. Donc $\frac{x_{n+1}-1}{x_n-1} = \frac{2}{3}$. La convergence est linéaire, avec $K_1 = \frac{2}{3}$. Ceci correspond à $1 - \frac{1}{m} = 1 - \frac{1}{3} = \frac{2}{3}$.