

Analyse numérique avec python

Yehor Korotenko

November 15, 2025

Abstract

Ce sont les notes prises aux CMs d'Analyse Numériques avec Python (UE MDD253DU) fait par le professeur Jean-Baptiste APOUNG KAMGA. Je n'ai rien ajouté dans ces notes. Ce qui est écrit dans ce fichier est ce que le professeur a écrit sur le tableau pendant les CMs.

Mes notes d'autres matières sont accessibles sur mon site: dobbikov.com

CONTENTS

| | |
|---|-----------|
| 1 Équations Différentielles | 3 |
| 1.1 Modèles discrètes | 3 |
| 1.1.1 Modèle de croissance géométrique | 3 |
| 1.2 Modèles continues | 4 |
| 1.2.1 Modèle de Malthus | 4 |
| 1.2.2 Modèle Verhulst | 5 |
| 1.3 Modèle de croissance logistique | 6 |
| 1.4 Notion de champ de vecteurs associée à une EDO | 6 |
| 1.4.1 Généralités et définitions | 6 |
| 1.4.2 Dessins de champs de vecteurs | 9 |
| 1.4.3 Recherche de solution approchée de modèles sous python | 9 |
| 1.5 Modèle de prédateur proie (lotka-volterra (1931)) | 10 |
| 2 Interpolation polynomiale | 11 |
| 2.1 Rappels sur les nnts numériques | |
| Vitesse (ordre) de convergence | |
| valeur ajoutée par itérations | 11 |
| 2.1.1 Valeur ajoutée par l'itération | 12 |
| 2.1.2 Obtenir numériquement la vitesse de convergence | 13 |
| 2.2 Interpolation: définition-motivation-exemples | 14 |
| 2.2.1 Définition | 14 |
| 2.2.2 Motivations | 14 |
| 2.2.3 exemples d'interpolation | 14 |
| 2.3 Polynôme interpolateur de lagrange | 17 |
| 2.3.1 Définition et propriétés | 17 |
| 2.3.2 Estimation d'erreur | 17 |
| 2.3.3 Implémentation avec python | 18 |
| 2.4 Construction des polynôme d'interpolation de lagrange | 18 |
| 2.4.1 Interpolation dans la base canonique (Vandermonde) | 18 |
| 2.4.2 Interpolation dans la base duale: Formule de lagrange et points barycentrique | 19 |
| 2.4.3 Méthode des différences divisées | 21 |
| 2.5 Comportement asymptotique "lorsque $N \rightarrow \infty$ " | 23 |
| 2.5.1 Observation | 23 |
| 2.5.2 Polynôme de Tchebychev | 23 |
| 2.5.3 Application | 24 |
| 3 Intégration numérique | 26 |
| 3.1 Formule de quadrature | 27 |
| 3.1.1 Construction de formule de quadrature à points donnés | 27 |
| 3.2 Utilisation des formules de quadrature. Formule de quadrature élémentaires et composées | 29 |
| 3.2.1 Définition | 29 |
| 3.2.2 Méthode de quadrature clémique | 30 |
| 3.2.3 Méthode de trapèze | 31 |
| 3.2.4 Méthode de point-milieu | 32 |
| 3.2.5 Méthode de Simpson | 33 |
| 3.2.6 Méthode de Newton-Côte | 34 |
| 3.3 Construction de formule de quadrature (à points inconnus): Formules de Gauss Legedre | 36 |

| | |
|---|-----------|
| 4 Résolution approchée d'équations différentielles ordinaires (EDOs) | 38 |
| 4.1 Motivations | 38 |
| 4.1.1 Définitions | 38 |
| 4.1.2 Exemple | 39 |
| 4.1.3 Nécessité de la solution approchée | 40 |
| 4.2 Problème d'évolution de population des lapins | 41 |
| 4.3 Exemple de Schémas numériques | 42 |
| 4.3.1 Formulation intégrale | 42 |
| 4.3.2 Construction de schéma d'Euler explicite | 42 |
| 4.3.3 Autres schémas et forme général des schemas explicites à un pas | 44 |
| 4.4 Étude de schémas pokes EDOs | 44 |
| 4.4.1 Définition | 44 |
| 4.4.2 Ordre d'un shéma à un pas explicite | 45 |
| 4.4.3 Stabilité des schémas à un pas | 47 |
| 4.4.4 Convergence des schémas à un pas explicite | 47 |
| 5 Résolution approchée d'équations ordinaires (EO) : $f(x) = 0$ | 49 |
| 5.1 Généralités et exemples | 49 |
| 5.1.1 Définitions | 49 |
| 5.1.2 Exemples | 49 |
| 5.2 Position correcte du problème (EO) | 50 |
| 5.3 Construction de schéma pour (EO) | 50 |
| 5.3.1 Méthode de dichotomie | 50 |
| 5.3.2 Méthode de fausse position | 52 |
| 5.4 Méthode d'itération (ou du point fixe) | 53 |
| 5.4.1 Principe | 53 |
| 5.4.2 Interpolation géométrique | 53 |
| 5.4.3 Algorithme | 53 |
| 5.4.4 Convergence | 54 |
| 5.5 Méthode de Newton | 55 |

CHAPTER 1

ÉQUATIONS DIFFÉRENTIELLES

1.1 Modèles discrètes

On désigne par $N(t)$ la population d'individus à l'instant t .

Équation du modèle discret:

$$\underbrace{N(t + \Delta t) - N(t)}_{\text{variation de la population}} = \underbrace{n}_{\text{nombre de naissances}} - \underbrace{m}_{\text{nombre de décès}} + \underbrace{i - e}_{\substack{\text{immigration} \\ \text{sol de migration}}} \quad \text{émigration}$$

1.1.1 Modèle de croissance géométrique

- hypothèse:

- solde migration nul: i.e $i - e = 0$
- nombre de croissance proportionnel à la taille de la population $n = \lambda \Delta t N(t)$
taux de natalité
- Idem pour le nombre de décès: $m = \mu \Delta t N(t)$ taux de mortalité

- Modèle: On pose $N_n = N(t_n)$ la taille de la population à l'instant t_n .

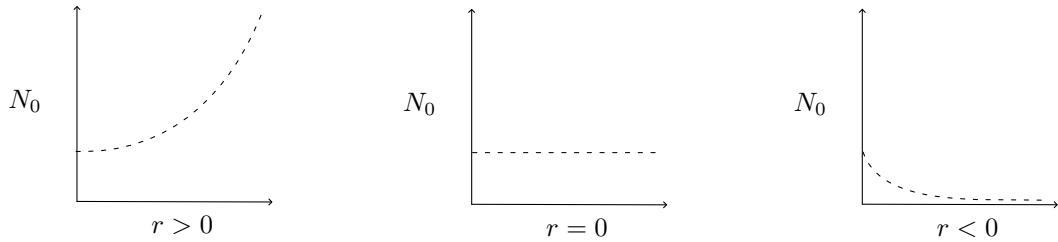
$$N_{n+1} - N_n = \lambda \Delta t N_n - \mu \delta t N_n$$

on pose $r = \lambda - \mu$

$$N_{n+1} = (1 + r \Delta t) N_n, \quad n = 0 \quad (1.1)$$

- Solution: $N_n = (1 + r \Delta t)^n N_0, \quad n \in \mathbb{N}$

- Visualisation: Δt fixé



(a) Natalité supérieure
à la mortalité

(b) Natalité égale
à la mortalité

(c) Natalité inférieure
à la mortalité

Property. .

- Lorsque $t \rightarrow 0$, la population semble tendre vers une courbe $N(t) = N_0 e^{rt}$, solution de $\begin{cases} N'(t) = rN(t) \\ N(0) = N_0 \end{cases}$
- Si $r > 0$, la population croît indéfiniment
- Si $r < 0$, il y a extinction de l'espèce.

Inconvénients:

1. Une croissance infinie n'est pas réaliste
2. Pour être rigoureux, on devrait écrire $E(rN_n)$ i.e partie entière.

1.2 Modèles continues

Motivation: L'observation qui prend Δt proche de 0 aura beaucoup plus d'information.

Remark 1.1. Le modèle de croissance géométrique

$$\begin{aligned} N(t + \Delta t) - N(t) &= \lambda \Delta t N(t) - \mu \Delta t N(t) \\ \Rightarrow \frac{N(t + \Delta t) - N(t)}{\Delta t} &= \lambda N(t) - \mu N(t) \end{aligned}$$

en faisant $\Delta t \rightarrow 0$

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

D'où l'équation des modèles continues:

$$\underbrace{N'(t)}_{\text{vitesse de variation}} = \underbrace{n(t)}_{\text{vitesse de naissance}} - \underbrace{m(t)}_{\text{vitesse de décès}} + \underbrace{i(t)}_{\text{vitesse d'immigration}} - \underbrace{e(t)}_{\text{vitesse d'émigration}}$$

1.2.1 Modèle de Malthus

- hypothèse:
 - solde migration nul: $i(t) - e(t) = 0$
 - vitesse de naissance proportionnel à la population à l'instant t : $n(t) = \lambda N(t)$
 - vitesse de décès: $m(t) = \mu N(t)$
- Modèle: $\begin{cases} N'(t) = (\lambda - \mu)N(t) \\ N(0) = N_0 \end{cases}$
- Solution: $N(t) = N_0 e^{(\lambda - \mu)t}$

Property. – Il peut être si comme limite du modèle de croissance géométrique.

- Lorsque $r = \lambda - \mu > 0$ croissance est proportionnel.
- Lorsque $r = \lambda - \mu = 0$ la population n'évolue pas.
- Lorsque $r = \lambda - \mu < 0$ la population tend vers 0.

- Inconvénients:

- croissance exponentielle pas réaliste. Il faut prendre en compte:
 - * la limitation des ressources
 - * l'interaction avec l'environnement

1.2.2 Modèle Verhulst

Corrige le modèle de Malthus en prenant en compte la limitation de ressources.

- Idée: limiter la croissance à un seuil K appelé capacité biotique

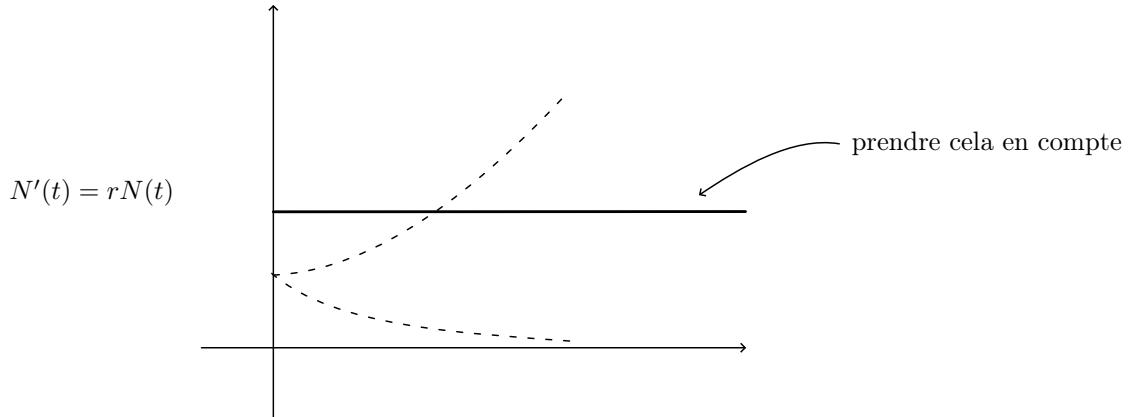


Figure 1.2: Modèle de Malthus

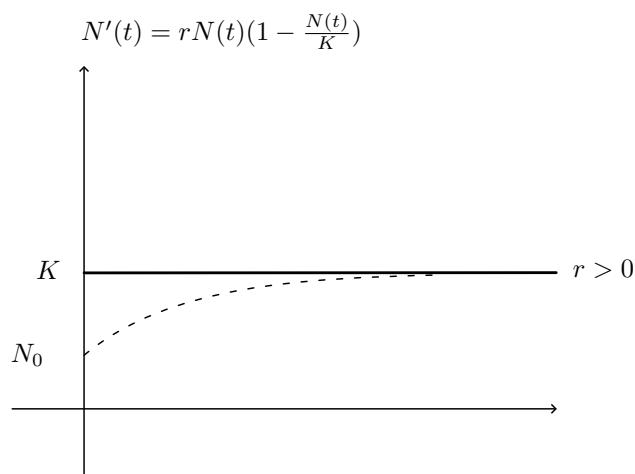


Figure 1.3: Modèle de Verhulst

- hypothèse: Sole de migration nul

- taux de natalité fonction affine décroissante de la population $\lambda \approx \lambda(1 - \frac{N(t)}{K})$
- taux de mortalité fonction affine croissante de la population $\mu \approx -\mu(1 - \frac{N(t)}{K})$

- Modèle: $\begin{cases} N'(t) = rN(t)(1 - \frac{N(t)}{K}) \\ N(0) = N_0 \end{cases}$

- Solutions: $N(t) = \frac{K}{1 + (\frac{K}{N_0} - 1)e^{-rt}}$ $t > 0$

- Visualisation:

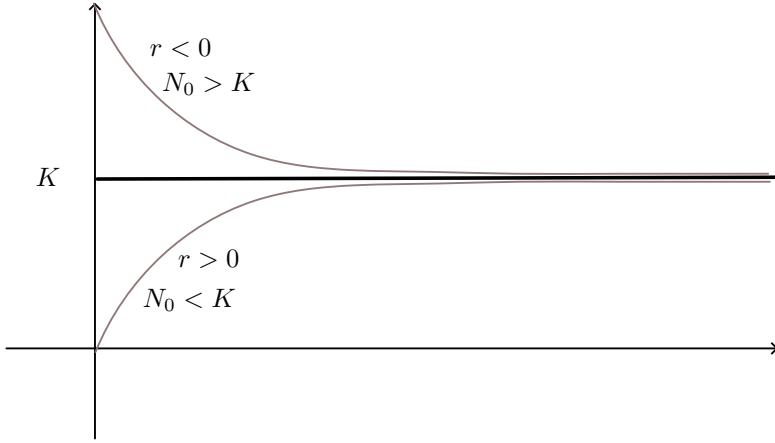


Figure 1.4: Verhulst solution

Property. Si $r > 0$, on a:

- si $N_0 = 0$ $N_0 = K$ on a: $N(t) = N_0 \forall t > 0$
- si $0 < N_0 < K$, N croissante
- si $N_0 > K$, N décroissante
- N possède une limite si $N_0 > 0$

$$\lim_{t \rightarrow \infty} N(t) = K$$

1.3 Modèle de croissance logistique

C'est un modèle discréte

- hypothèse: i.e $= 0$
 $n - m$ est une fonction affine de la population, i.e $n - m = r\Delta t N(t)(1 - \frac{N(t)}{K})$
- Modèle: On suppose $\Delta t = 1$: On pose $N_n = N(t_n)$

$$\text{On a: } \begin{cases} N_{n+1} - N_n = rN_n(1 - \frac{N_n}{K}) \\ N_0 \text{ donné} \end{cases}$$

Property. (À vérifier numériquement)

- si $r < 2$, la suite converge vers K
- si $2 < r < 2.449$, la suite converge vers un cycle
- si $2.449 < r < 2.57$, la suite est encore un cycle mais plus complexe
- si $r > 2.57$, la suite devient chaotique

1.4 Notion de champ de vecteurs associée à une EDO

1.4.1 Généralités et définitions

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

$$(EDO) \begin{cases} y'(x) = f(t, y(t)) & t \in]0, \pi[\\ y(0) = y_0 \end{cases}$$

Où

$$\begin{aligned} y : [0, \pi] &\longrightarrow \mathbb{R} \\ t &\longmapsto y(t). \end{aligned}$$

$$f :]0, \pi[\times \mathbb{R} \longrightarrow \mathbb{R}$$

$$(t, x) \longmapsto f(t, x).$$

- Si l'on sait résoudre analytiquement l'EDO (i.e donner l'expression de $t \mapsto y(y)$) alors c'est terminé car il suffit d'étudier la fonction $t \mapsto y(t)$
- Si l'on ne sait pas déterminer la solution analytique, on peut:
 1. s'assurer de **l'existence** et **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 pour simple analyse de $f(t, x)$

C'est ici qu'intervient les champs de vecteurs.

Illustrations.

1. Prenons le modèle de Malthus

$$\begin{cases} N'(t) = rN(t), & t \in]0, \pi[\\ N(0) = N_0 \end{cases}$$

On sait que $N(t) = N_0 e^{rt}$

2. Voici ce que fait python pour traiter N .

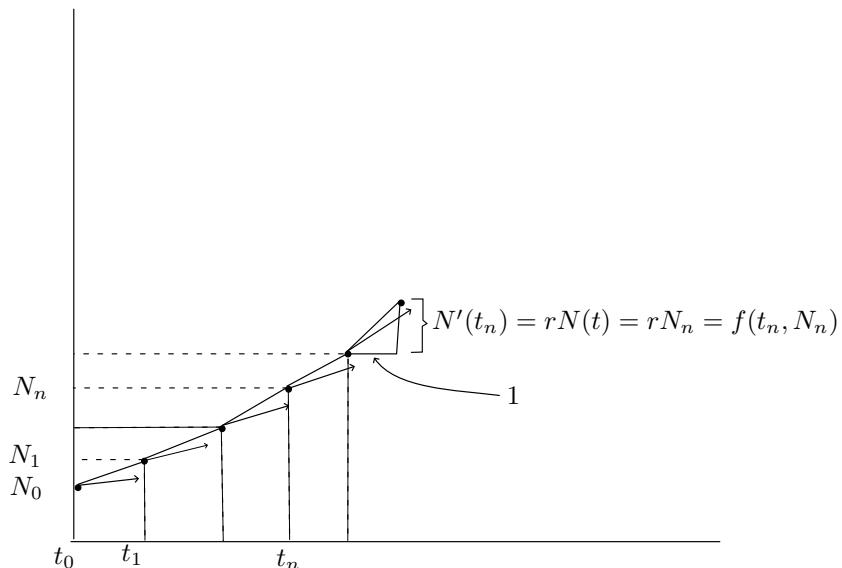


Figure 1.5: Ce que fait python

3. Traitons les vecteurs tangents à la courbe $t \mapsto N(t)$ aux points t_n , $n = 0$
4. Si l'on connaît les valeurs minimales et maximales de la solutions on peut avoir l'allure de la solution.

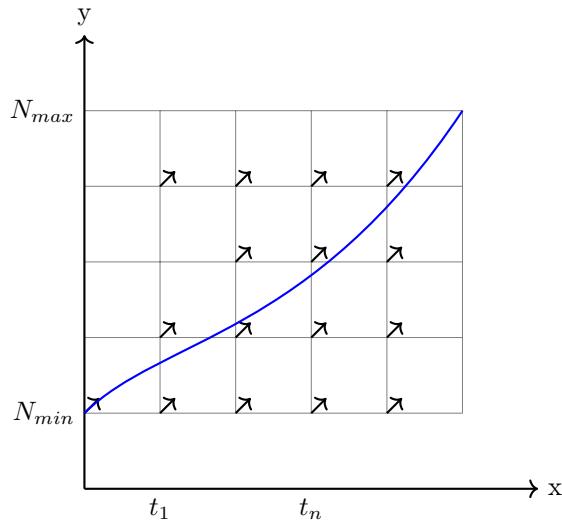


Figure 1.6: Une courbe sur des champs de vecteurs

Analysons ce que représente le vecteur tangent:

- pour une courbe $y = g(x)$
- python et tout autre logiciel procède ainsi

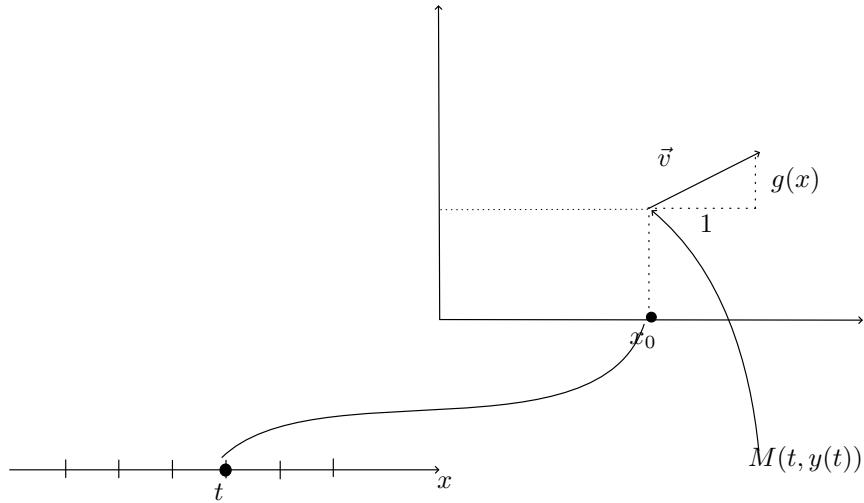


Figure 1.7: Ce que représente vecteur

Le vecteur tangent à la courbe:

$$\begin{aligned}\vec{v} &= (1, g'(x)) = \left(1, \frac{dy}{dx}\right) = \left(1, \frac{\frac{dy}{dt}}{\frac{dx}{dt}}\right) \\ &= \frac{1}{\frac{dx}{dt}} \left(\frac{dx}{dt}, \frac{dy}{dt}\right) = \frac{1}{\dot{x}(t)} \underbrace{(\dot{x}(t), \dot{y}(t))}_{\in \mathbb{R} \text{ vecteur tangent}} \\ \vec{v} &= (\dot{x}(t), \dot{y}(t))\end{aligned}$$

Càd \vec{v} est le vecteur vitesse au points $M(x(t), y(t))$ à la courbe paramétrée $t \mapsto \begin{cases} x(t) = t \\ y(t) = g(t) \end{cases}$. On a le résultat.

Proposition 1.2.

(y obtient solution de l'EDO $y'(t) = f(t, y(t))$)

\Updownarrow

(vecteur vitesse de la courbe paramétrée $t \mapsto (x(t), y(t))$ au point $M(t_0) = (t_0, y(t_0))$
si le vecteur $(1, f(t_0, y(t_0)))$)

Proposition 1.3.

$$V : \mathbb{R}^2 \longrightarrow \mathbb{R}^2 \\ (t, y) \longmapsto V((t, y)).$$

(si le champ de vecteur associé à l'EDO $y'(t) = f(t, y(t)) \Leftrightarrow V(t, y) = (1, f(t, y))$)

1.4.2 Dessins de champs de vecteurs

Principe:

À chaque points $P = (p_x, p_y)$ on trace le vecteur $\varepsilon V(P)$ où ε est une constance positive choisi pour écrire les vecteurs trop longs.

Avec python on écrit `quiver(Px, Py, Vx, Vy, angles='xy')` RQ 1: Cette fonction est vectorielle, i.e P_x, P_y, V_x, V_y , sont des numpy array de taille n . RQ 2: On peut ajouter un paramètre pour controles la longueur des vecteurs:

```
plt.quiver(Px, Py, Vx, Vy, angles='xy', scale=1)
```

Par conséquent, il faut normaliser les vecteurs (i.e le champ de vecteur)

Example 1.4. Champ de vecteur du modèle de Verhulst:

```
1 def f(t, y):
2     return r * y * (1 - y/k)
```

la grille:

```
1 lt = np.linspace(tmin, tmax, N+1)
2 ly = np.linspace(ymin, ymax, M+1)
3 T, Y = np.meshgrid(lt, ly)
```

Construire les vecteurs:

```
1 Y = 1 + 0 * T
2 V = f(T, Y)
3 norm = np.sqrt(U*U + V*V)
4 U = U/norm
5 V = V/norm
```

On place les points:

```
1 plt.scatter(T, Y, marker='+', alpha = 0.5)
```

On place les vecteurs

```
1 plt.quiver(T, Y, U, V, angles='xy', scale=N)
```

1.4.3 Recherche de solution approchée de modèles sous python

On cherche une solution approchée de

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

avec python. Pour cela il suffit de dire **en quels points** on veut cette solution.
On se donne:

- une liste des instants $[t_0, t_1, \dots, t_N]$
- t_0, y_0
- Puis, on appelle la fonction `odeint` du module `scipy.integrate` de python.
- On obtient une liste $[y_0, y_1, \dots, y_N]$

Example 1.5. Cas du modèle du Verhulst

- EDO:

```
1     def f(t, y):
2         return \ldots
```

- Instants

```
1     t0, tf = a, b
2     N = 100
3     t = np.linspace(t0, tf, N)
```

- On appelle `odeint`

```
1     from scipy.integrate import odeint
2     yapp = odeint(f, t, y), rtol=None, atol=None, tfloat=False)
3     plt.plot(t, yapp, \ldots)
```

1.5 Modèle de prédateur proie (lotka-volterra (1931))

$H(t)$: population de sardins

$P(t)$: population de requins

$$\frac{H'(t)}{H(t)} = \text{taux de variation de sardins} = \underbrace{a}_{\text{taux de croissance}} - \underbrace{bP(t)}_{\text{taux de mortalité}}$$

$$\frac{P'(t)}{P(t)} = \text{taux d'arrivé des requêtes} = \underbrace{-c}_{\text{taux de décès}} + \underbrace{dH(t)}_{\text{taux de croissance}}$$

D'où le modèle:

$$\begin{cases} H'(t) = H(t)(a - bP(t)) & t > 0 \\ P'(t) = P(t)(-c + dH(t)) \\ H(0) = H_0, \quad P(0) = P_0 \end{cases}$$

Si l'on désigne par $p \geq 0$ la proportion des requêtes en sardines pêchées

$$\begin{cases} H'(t) = H(t)(a - p - bP(t)) & t > 0 \\ P'(t) = P(t)(-c - p - dH(t)) \\ H(0) = H_0 \\ P(0) = P_0 \end{cases}$$

CHAPTER 2

INTERPOLATION POLYNOMIALE

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

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

- savoir si le polynôme construit est d'autant plus proche de la fonction que le nombre de point est grand. C'est-à-dire, est-ce que nute des "erreurs" tend vers zero lorsque le nombre de points tend vers l'infini.
- Si oui, comment quantifier cette convergence? C'est-à-dire, quelle est la vitesse (ordre) de cette convergence.

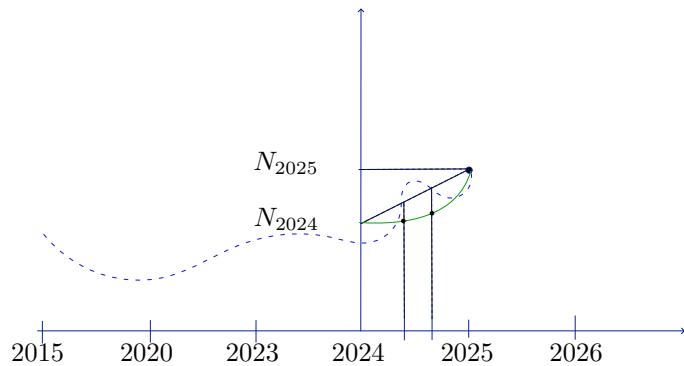


Figure 2.1: evolution-de-population-en-annee

1. Approche 1: approximation linéaire.
 - Polynôme de degré 1
2. Approche 2:
 - polynôme de degré 2
 - approximation quadratique
3. Approche 3: prise en compte d'Historique

2.1 Rappels sur les nuts numériques Vitesse (ordre) de convergence valeur ajoutée par itérations

Definition 2.1. Soit $(x_n)_n \subset \mathbb{R}^n$ une suite qui converge vers $x^* \in \mathbb{R}^n$, pour une norme $\|\cdot\|$ de \mathbb{R}^n

- Si $k_1 = \lim_{n \rightarrow \infty} \frac{\|x_{n+1} - x^*\|}{\|x_n - x^*\|}$ existe et $k_1 \in]-1, 1[\setminus \{0\}$. On dit que la suite converge linéairement vers x^* ou que la convergence est d'ordre 1.
- 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.
- 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 . La constante K_q est appelée constante asymptotique d'erreur.

Example 2.2. 1. $x_n = (0.2)^n$

- On a $\lim_{n \rightarrow \infty} x_n = 0$. La convergence vers $x^* = 0$.
- $\lim_{n \rightarrow \infty} \frac{|x_{n+1} - x^*|}{|x_n - x^*|} = \lim_{n \rightarrow \infty} \frac{(0.2)^{n+1}}{(0.2)^n} = 0.2 \in]-1, 1[\setminus \{0\}$

D'où

- x_n converge à l'ordre 1
- Sa constante asymptotique est $k_1 = 0.2$

2. $I_n = (0.2)^{2^n}$. On a $\lim_{n \rightarrow \infty} I_n = 0$

On a:

$$\begin{aligned} I_{n+1} &= (0.2)^{2^{n+1}} = (0.2)^{2^{n+2}} \\ &= ((0.2)^{2^n})^2 \\ &= (I_n)^2 \end{aligned}$$

D'où $\lim_{n \rightarrow \infty} \frac{I_{n+1}}{(I_n)^2} = \lim_{n \rightarrow \infty} \frac{(I_n)^2}{(I_n)^2} = 1$ D'où

- convergence d'ordre 2
- de constante $k_2 = 1$

En pratique, on ne dispose pas de K_q

Definition 2.3.

$$x_n \text{ converge vers } x^* \Rightarrow \exists N \in \mathbb{N}, \exists A, B \in \mathbb{R} \text{ tq } \forall n \geq N, 0 < A \leq \frac{\|x_{n+1} - x^*\|}{\|x_n - x^*\|^q} \leq B < +\infty$$

La convergence est au moins d'ordre q si et seulement si on a (deuxième partie d'équation)

2.1.1 Valeur ajoutée par l'itération

Il est question de comparer 2 suites qui ont la même vitesse de convergence.

Remark 2.4. Si $|x_n - x^*| = 4 \cdot 10^{-8} = 0.\underbrace{0000000}_{7 \text{ chiffres}} 4$. On dira que x_n et x^* ont 7 chiffres exactes après la virgule.

$$\begin{aligned} \log_{10} |x_n - x^*| &= \log_{10} 4 - 8 \log_{10}(10) \\ \frac{\log |x_n - x^*|}{\log 10} &= \frac{\log 4}{\log 10} - 8 \end{aligned}$$

i.e $d_n = -\log_{10} |x_n - x^*|$ mesure le nombre de chiffres décimales entre x_n et x^* qui coïncident.

Remark 2.5.

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

D'où $d_{n+1} - qd_n \approx -\log_{10} K_q$, i.e

$$d_{n+1} + \frac{\log_{10} K_q}{1-q} \approx q(d_n + \frac{\log_{10} K_q}{1-q})$$

Donc, le nombre de chiffres significatifs est multiplié par q .

Proposition 2.6. Si x_n converge à l'ordre 1 vers x^* de constante asymptotique K_1 , alors le nombre d'itérations nécessaires pour gagner un chiffre exacte est la partie entière de $-\frac{1}{\log_{10} K_1}$

Proof. Soit m le nombre d'itérations pour gagner un chiffre. Comme $d_{n+1} - d_n = -\log_{10} K_1$, en partant de d_n , après m itérations on aura

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

D'où on aura gagné 1 chiffre si $d_{n+m} - d_n = 1$, i.e

$$1 = -m \log_{10} K_1 \Rightarrow m = \left(-\frac{1}{\log_{10} K_1} \right)$$

□

2.1.2 Obtenir numériquement la vitesse de convergence

On cherche q tq: $\lim_{n \rightarrow \infty} \frac{\|x_{n+1} - x^*\|}{\|x_n - x^*\|^q} = K_q \in \mathbb{R}^*$

Remark 2.7.

$$\frac{\|x_{n+1} - x^*\|}{\|x_n - x^*\|^q} \approx K_q \Rightarrow \underbrace{\log \|x_{n+1} - x^*\|}_Y - \underbrace{q \log \|x_n - x^*\|}_X = \log K_q$$

i.e $Y = aX + b$.

Conclusion: pour déterminer q :

- Traiter la courbe $\log \|x_n - x^*\| \mapsto \log \|x_{n+1} - x^*\|$
- Déterminer q comme la pente de la droite passant par le maximum de points.

$$\begin{aligned} x_n &= x_0, x_1, \dots, x_N \\ x_n - x^* &= x_0 - x^*, x_1 - x^*, \dots, x_N - x^* \\ x_{n+1} - x^* &= x_1 - x^*, x_2 - x^*, \dots, x_{N+1} - x^* \end{aligned}$$

En python:

```

1 xn = np.array([x0, ..., xN])
2 e = np.log(np.abs(xn - x^*))
3 ex = e[0:-1] # de premier à avant dernier
4 ey = e[1:] # de deuxième au dernier
5 plt.scatter(ex, ey, label="image")
6 a,b = np.polyfit(ex, ey, 1)
7 plt.plot(ex, b + a * ex, label=f"$y = {b:.32f} + {a:.32f}x$")

```

2.2 Interpolation: définition-motivation-exemples

2.2.1 Définition

Definition 2.8. Soient $(x_i, y_i)_{i=\{1, \dots, N\}}$ un nuage de points (exemple un ensemble discret de point du graphe d'une fonction). Interpoler ce nuage de points correspond à chercher un polynôme de degré $N - 1$, qui passe par chacun de ces points.

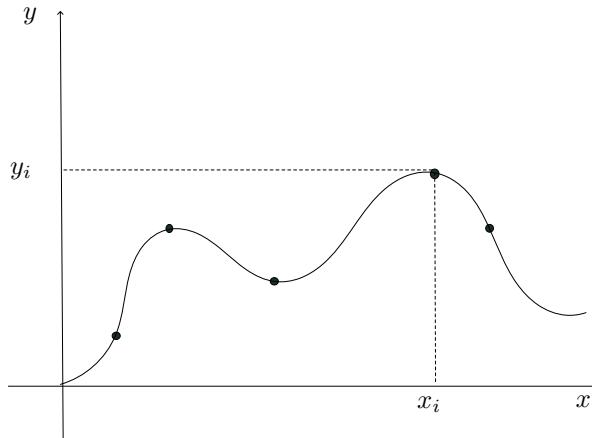


Figure 2.2: L'exemple visuel de la définition

Questions:

1. Comment le construire?
2. $P_{N-1} \in \mathbb{R}_{N-1}[X]$
3. $P_{N-1}(x_i) = y_i$

2.2.2 Motivations

- La solution d'un problème est fournie par une formule représentative: Noyau de la chaleur (i.e convolution) est un cherche la solution en un nombre de points.
 - 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 table des valeurs en un nombre fini de points et on souhaite l'évaluer partout.
 - l'interpolation
- On peut utiliser l'interpolation dans
 - l'intégration numérique
 - la résolution numérique des EDO
 - la visualisation scientifique

Definition 2.9. Un tel polynôme est appelé **polynôme interpolateur de lagrange** de degré $N - 1$ de ces points.

2.2.3 exemples d'interpolation

Theorem 2.10. Polynôme intérpolateur de degré 1.

Soient $(x_1, y_1), (x_2, y_2)$ 2 points distincts de \mathbb{R}^2

- Il existe une unique droite passant par les 2 points.

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

- Si de plus, $x_1 \neq x_2$, il existe un unique polynôme de degré 1 (i.e $P_1 \in \mathbb{R}_1[X]$) tq:

$$(x, y) \in \mathcal{D} \Leftrightarrow y = P(x) \text{ avec } P_1 = \frac{(x - x_1)y_1 - (x - x_2)y_2}{x_2 - x_1}$$

Proof.

•

$$\begin{aligned} M \begin{pmatrix} x \\ y \end{pmatrix} \in \mathcal{D} &\Leftrightarrow M \vec{M}_1 / M_1 \vec{M}_2 \\ &\Leftrightarrow \det(M \vec{M}_1, M_1 \vec{M}_2) = 0 \\ &\Leftrightarrow \begin{vmatrix} x - x_1 & x_2 - x_1 \\ y - y_1 & y_2 - y_1 \end{vmatrix} = 0 \\ &\Leftrightarrow (x - x_1)(y_2 - y_1) - (y - y_1)(x_2 - x_1) = 0 \end{aligned}$$

- Si $x_1 \neq x_2$

$$\begin{aligned} M \in \mathcal{D} &\Leftrightarrow y - y_1 = \frac{(x - x_1)(y_2 - y_1)}{x_2 - x_1} \\ &\Leftrightarrow y = P_1(X) \end{aligned}$$

□

Remark 2.11. On a l'écriture équivalente de P_1 :

•

$$P_1 \frac{x_0 y_1 - x_1 y_2}{x_2 - x_1} + X \frac{y_2 - y_1}{x_2 - x_1} \equiv a_0 + a_1 X$$

C'est l'écriture dans la base $(1, X)$ de $\mathbb{R}_1[X]$

•

$$P_1 = \underbrace{\frac{x - x_2}{x_1 - x_2}}_{l_1} y_1 + \underbrace{\frac{x - x_1}{x_2 - x_1}}_{l_2} y_2$$

C'est l'écriture dans la base (l_1, l_2) de $\mathbb{R}_1[X]$

RQ:

$$l_1(x_1) = 1 \quad l_1(x_2) = 0 \quad l_2(x_1) = 0 \quad l_2(x_2) = 1$$

(base de lagrange)

•

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

C'est l'écriture dans la base $(1, x - x_1)$ de $\mathbb{R}_1[X]$ (base de newton)

Example 2.12. 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: $x_1 \neq x_2 \neq x_3$

P_2 est un polynôme de degré 2

$$P_2 = a_0 + a_1x + a_2x^2$$

Lemma 2.13.

$$P_2(x_1) = y_1, \quad P_2(x_2) = y_2 \quad \text{i.e } a_0 + a_1x_1 + a_2x_1^2 = y_1$$

$$\underbrace{\begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{bmatrix}}_M \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}}_{\{ } = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}}_{\{ } \Rightarrow \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}}_{\{ } = \underbrace{M^{-1}}_{\{ \text{??}} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

Remark 2.14. Par 2 points

$$M = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \end{bmatrix} \text{ et } M^{-1} = \frac{1}{x_2 - x_1} \begin{bmatrix} x_2 & -x_1 \\ -1 & 1 \end{bmatrix}$$

- Méthode 2:

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

$$\begin{aligned} P_2(x_i) = y_i &\Rightarrow \begin{cases} a_0 &= y_1 \\ a_0 + a_1(x_2 - x_1) &= y_2 \\ a_0 + a_1(x_3 - x_1) + a_2(x_3 - x_1)(x_3 - x_2) &= y_3 \end{cases} \\ &\Rightarrow \begin{cases} a_0 &= y_1 \\ a_1 &= \frac{y_2 - y_1}{x_2 - x_1} \\ a_2 &= \frac{y_3 - y_1 - \frac{y_2 - y_1}{x_2 - x_1}(x_3 - x_1)}{(x_3 - x_1)(x_3 - x_2)} \end{cases} \end{aligned}$$

Remark 2.15. On a:

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

càd

$$P_3 = y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_1) + \frac{\frac{y_2 - y_1}{x_2 - x_1} - \frac{y_3 - y_2}{x_3 - x_2}}{x_3 - x_2}(x - x_1)(x - x_2)$$

| | | | |
|-------|--------------|--------------------------------------|--|
| x_1 | $y_1 =: a_3$ | | |
| x_2 | y_2 | $\frac{y_2 - y_1}{x_2 - x_1} =: a_1$ | |
| x_3 | y_3 | $\frac{y_3 - y_2}{x_3 - x_2}$ | $\frac{\frac{y_2 - y_1}{x_2 - x_1} - \frac{y_3 - y_2}{x_3 - x_2}}{x_3 - x_1} =: a_2$ |

- Méthode 3:

$$P_3 = \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)}y_1 + \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)}y_2 + \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)}y_3 = \sum_{i=1}^3 \underbrace{\left(\prod_{j=1}^3 \frac{x - x_j}{x_i - x_j} \right)}_{l_i(x)} y_i$$

Remark 2.16. $(x_1, y_1), (x_2, y_2)$

$$y = \frac{x - x_2}{x_1 - x_2} y_1 + \frac{x - x_1}{x_2 - x_1} y_2$$

2.3 Polynôme interpolateur de lagrange

2.3.1 Définition et propriétés

Theorem 2.17. (existence et utilité)

Soit x_1, \dots, x_n des réels 2 à 2 distincts et y_1, \dots, y_n des réels quelconques: Il existe un unique polynôme $P \in \mathbb{R}_{n-1}[X]$ (i.e de degré $n-1$) tel que $p(x_i) = y_i, i = 1, \dots, n$

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

Proof. Soit

$$\begin{aligned}\Phi : \mathbb{R}_{n-1}[X] &\longrightarrow \mathbb{R}^{n-1} \\ P &\longmapsto \Phi(P) = (P(x_1), \dots, P(x_n)).\end{aligned}$$

on a:

- Φ linéaire
- Φ injective

En effet, $\Phi(P) = 0 \Leftrightarrow P(x_i) = 0 \Leftrightarrow P \equiv 0$ car $\deg(P) \leq n-1$. D'où Φ isomorphisme d'espace vectoriel et la surjection assure le résultat. \square

Definition 2.18. Si f est continue sur $[a, b] \rightarrow \mathbb{R}$, $x_1, \dots, x_n \in [a, b]$ 2 à 2 distincts, alors, l'unique $P \in \mathbb{R}_{n-1}[X]$ tq $P(x_i) = f(x_i) i = 1, \dots, n$ est appelé polynôme d'interpolation de lagrange de f aux points x_1, \dots, x_n

2.3.2 Estimation d'erreur

Theorem 2.19. l'erreur

Soient

- $a < b$ $f : [a, b] \rightarrow \mathbb{R}$ continue
- x_1, \dots, x_n 2 à 2 distincts de $[a, b]$
- P polyôme d'interpolation de lagrange de f aux points x_i

Si f est n fois dérivable sur $]a, b[$, alors, pour tout $a \in [a, b]$, il existe $t \in]a, b[$ tq

$$f(x) - P(x) = \omega_n(x) \frac{f^{(n)}(t)}{n!}$$

où $\omega_n(x) = (x - x_1) \dots (x - x_n)$

Corollary 2.20. Si $f^{(n)}$ est bornée par M sur $]a, b[$, alors $\forall x \in [a, b]$

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

Proof. à faire □

2.3.3 Implémentation avec python

```

1  from scipy.interpolate import lagrange
2  x = np.array([x_1, x_2, x_3])
3  y = np.array([y_1, y_2, y_3])
4  p = lagrange(x, y)
5  print(p) # affiche le polynôme
6  print(p(3)) # callable

```

2.4 Construction des polynômes d'interpolation de Lagrange

x_0, \dots, x_{n-1} 2 à 2 distincts

2.4.1 Interpolation dans la base canonique (Vandermonde)

Construction

$$P = \sum_{i=1}^{n-1} a_i x^i$$

$$P(x_i) = y_i, \quad i = 0, \dots, n-1$$

$$\underbrace{\begin{bmatrix} 1 & x_0 & \dots & x_0^{n-1} \\ 1 & x_1 & \dots & x_1^{n-1} \\ \dots & \dots & \dots & \dots \\ 1 & x_{n-1} & \dots & x_{n-1}^{n-1} \end{bmatrix}}_{V(x_0, \dots, x_{n-1})} \underbrace{\begin{bmatrix} a_0 \\ \dots \\ a_{n-1} \end{bmatrix}}_a = \underbrace{\begin{bmatrix} y_1 \\ \dots \\ y_{n-1} \end{bmatrix}}_b$$

Matrice de Vandermonde

- elle pleine
- malconditionnée

```

1 def VDM_Mat(x):
2     x_n = np.reshape(x, (x.size, 1))
3     return x_n ** np.arange(x.size)

```

```

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

```

Evaluation efficace de P algorithme de Horner

Proposition 2.21. Si X est un réel et Q est le polynôme défini par

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

alors la suite

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

vérifie $q_n = Q(x)$

Proof. (laissé exo)

$$Q(X) = X^2 + 2X + 1 \equiv (X + 2)X + 1$$

□

```

1 def Horner(P, xx):
2     y = 0
3     for a in P:
4         y = y * xx + a
5     return y

```

```

1 def IntuP_VDM(x, y, xx):
2     a = VDM_Poly(x, y)
3     yy = Horner(a[::-1], xx)
4     return yy

```

2.4.2 Interpolation dans la base duale: Formule de lagrange et points barycentrique

Construction

Idée prendre pour base de $\mathbb{R}_{n-1}[X]$ l'image réciproque de la base canonique de \mathbb{R}^{n-1} pour Φ

$$\begin{aligned} L_i(x_j) &= \begin{cases} 1 & \text{si } i \neq j \\ 0 & \text{sinon} \end{cases} \\ L_i(x) &= \frac{\prod_{j \neq i}^{n-1} (x - x_j)}{\prod_{j=0}^{n-1} (x_i - x_j)} = \prod_{j=1, j \neq i}^{n-1} \frac{x - x_j}{x_i - x_j} \\ P(X) &= \sum_{i=0}^{n-1} y_i L_i(X) \end{aligned}$$

Theorem 2.22.

$$\begin{aligned} f : [a, b] &\rightarrow \mathbb{R} \\ x_1, \dots, x_n \\ P \end{aligned}$$

Si f n fois dérivable,

$$\forall x \in [a, b], \exists t \in]a, b[, f(x) - P(x) = \omega_n(x) \frac{f^{(n)}(t)}{n!}$$

Proof. du théorème (erreur)

Soit x fixé des $[a, b] \setminus \{x_1, \dots, x_n\}$

On pose

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

F est n fois dérivable et P annule aux $n+1$ points x_1, \dots, x_n, x . D'après le théorème Rolle (généralisé)

$$\exists t \in]a, b[\text{ tq } f^{(n)}(t) = 0$$

Or

$$\underbrace{F^{(n)}(t)}_{=0 \text{ par hyp}} = f^{(n)}(t) - \underbrace{P^{(n)}(t)}_{=0 \text{ car } \deg(P) < n} - \frac{f(x) - P(x)}{\omega_n(x)} n!$$

D'où

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

Par ailleurs, si $x \in \{x_1, \dots, x_n\}$, $f(x) - P(x) = 0$

$$\omega_n(x) = (x - x_1) \dots (x - x_n)$$

□

Proof. corollaire

$$|f(x) - P(x)| = |\omega_n(x)| \frac{|f^{(n)}(t)|}{n!}$$

comme $x, x_i \in [a, b]$, on a $|x - x_i| \leq b - a$ et $|f^{(n)}(t)| \leq M$, on a:

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

□

Evaluation efficace: formule barycentrique

Proposition 2.23. On a

$$\begin{aligned} P(x) &= \sum_{i=1}^n y_i \frac{\omega_n(x)}{(x - x_i)\omega'_n(x_i)} \\ &= \frac{\sum_{i=1}^n \frac{1}{(x - x_i)\omega'_n(x_i)} y_i}{\sum_{i=1}^n \frac{1}{(x - x_i)\omega'_n(x_i)}} \end{aligned}$$

Proof. Comme

$$\omega_n(x) = \prod_{i=1}^n (x - x_i) \Rightarrow \omega'_n(x) = \sum_{i=1}^n \prod_{j=1, j \neq i}^n (x - x_j)$$

D'où

$$\omega'_n(x_i) = \prod_{j=1, j \neq i}^n (x_i - x_j) \quad i = 1, \dots, n$$

D'où

$$L_i(x) = \frac{\omega_n(x)}{x - x_i} \frac{1}{\omega'_n(x_i)}$$

Et

$$\begin{aligned} \sum_{i=1}^n y_i L_i(x) &= \sum_{i=1}^n \frac{y_i}{(x - x_i)\omega'_n(x)} \omega_n(x) \\ &= \omega_n(x) \sum_{i=1}^n \frac{y_i}{(x - x_i)\omega'_n(x_i)} \end{aligned}$$

Or pour $P \equiv 1$ on a $y_i = 1, i = 1, \dots, n$, on a

$$1 = \omega_n(x) \sum_{i=1}^n \frac{1}{(x - x_i)\omega'_n(x)}$$

D'où

$$\omega_n(x) = \left(\sum_{i=1}^n \frac{1}{(x - x_i)\omega'_n(x)} \right)^{-1}$$

Enfin,

$$P(x) = \frac{\sum_{i=1}^n \frac{y_i}{(x-x_i)\omega'_n(x)}}{\sum_{i=1}^n \frac{1}{(x-x_i)\omega'_n(x)}}$$

□

Remark 2.24. 1. Attention: si $x = x_i$, $i = 1, \dots, n$

- 2. Exercice: calculer la complexité de cette formule et comparer à la première.
- 3. Ajouter un nouveau point d'interpolation oblige à refaire tous les calculs.

2.4.3 Méthode des différences divisées

Préliminaires: Interpolation de Neville

Lemma 2.25. Considérons n points 2 à 2 distincts x_1, \dots, x_n et n réels y_1, \dots, y_n . Pour $1 \leq k \leq l \leq n$, posons P_{x_k, \dots, x_l} le polynôme d'interpolation aux points

$$(x_k, y_k) \dots (x_l, y_l)$$

Nous avons

$$P_{x_k, \dots, x_l}(x) = \frac{(x - x_l)P_{x_l, \dots, x_{l-1}}(x) - (x - x_k)P_{x_{k+1}, \dots, x_l}(x)}{x_k - x_l}$$

Schématiquement

$$\begin{aligned} P(x) &= x_k, \underbrace{x_{k+1}, \dots, x_{l-1}}_{P_1}, x_l \\ &\quad \frac{x - x_l}{x_k - x_l} P_1 + \frac{x - x_k}{x_l - x_k} P_2 \end{aligned}$$

Construction de l'interpolation de Newton

Definition 2.26. (Polynôme de Newton) Soit $n \geq 1$ entier, x_1, \dots, x_n n réels 2 à 2 distincts. Les polynômes de Newton $\omega_0, \dots, \omega_n$ associés à ces points sont définis par

$$\begin{cases} \omega_0 = 1 \\ \omega_j = (x - x_1) \dots (x - x_j), \quad (1 \leq j \leq n) \end{cases}$$

Remark 2.27. $\{\omega_j\}_{j=1, \dots, k}$ est une base de $\mathbb{R}_k[x]$

- Ainsi le polynôme d'interpolation de Lagrange associé aux points $(x_1, y_1) \dots (x_n, y_n)$ s'écrit

$$P = \sum_{k=0}^{n-1} \alpha_k \omega_k$$

où α_k sont solutions de

$$y_i = \sum_{k=0}^{n-1} \alpha_k \omega_k(x_i), \quad i = 1, \dots, n$$

On parle de développement de Newton du polyôme de Lagrange

Definition 2.28. On appelle différences divisées d'ordre $j-1$ ($1 \leq j \leq n$) associées aux points $(x_1, y_1), \dots, (x_i, y_i)$ les nombres $d_{i,j}$ ($i = j$ à n) définis par

- $d_{i,1} = y_i \quad i = 1, \dots, n$
- $d_{i,j} = \frac{d_{i,j-1} - d_{i-1,j-1}}{x_i - x_{i-j+1}} \quad j = 2$ à n , $i = j$ à n

Lorsque $y_i = f(x_i) \quad i = 1, \dots, n$, $d_{i,j}$ est généralement noté $f[x_{j-i+1}, \dots, x_{j-1}, x_j]$ et est appelé différence divisée d'ordre $j-1$ aux i points x_{j-i+1}, \dots, x_j

Python:

```

1 def MatriceDifferencesDivisee(x, y):
2     n = len(y)
3     d = np.zeros((n, n))
4     d[:, 0] = 1.0 * y
5     for j in range(1, n):
6         d[j:n, j] = (d[j:n, j-1] - d[j-1:n, j-1]) / (x[j:n] - x[0:n-j])
7     return d

```

Remark 2.29. • Le "stencil" (squelette) est

$$\begin{array}{c} i-1, j-1 \\ \diagup \\ i, j-1 \text{ --- } i, j \end{array}$$

- La hauteur de stencil est j
- Le support du stencil est $[x_{i-j}, \dots, x_i]$

Proposition 2.30. On a $d_{j,j} = \alpha_{j-1}$ pour $j \in [1, \dots, n]$, càd:

$$P = \sum_{j=1}^n d_{j,j} \omega_{j-1}$$

Ainsi, pour calculer P il suffit de connaître $d_{j,j} \quad j = 1, \dots, n$

Calcul efficace du polynôme

Proposition 2.31. Soit donné x_0, \dots, x_n dés réels 2 à 2 distincts. Soit Q le polyôme défini par

$$Q(x) = a_0 + \sum_{i=1}^n a_i \prod_{j=1}^{i-1} (x - x_j) \equiv \sum_{i=0}^n a_i \omega_i(x)$$

La suite des polynômes Q_0, \dots, Q_n définies par

$$\begin{cases} Q_n = a_n \\ Q_k = a_k + (x - x_k)Q_{k-1} \quad k = n-1, \dots, 0 \end{cases}$$

vérifie $Q_0 = Q$

```

1 def HornerNewton(d, x, xx):
2     n = len(d)
3     yy = 0 * xx + d[n-1]
4     for i in range(n-2, -1, -1):
5         yy = d[i] + (xx - x[i]) * yy

```

```

6     return yy

1 def DifferencesDivisees(x, y):
2     d = MatriceDifferencesDivisee(x, y)
3     a = np.diag(d)
4     return a

```

2.5 Comportement asymptotique "lorsque $N \rightarrow \infty$ "

2.5.1 Observation

On n'a pas toujours une convergence uniforme de l'interpolation

Example 2.32. $f(x) = \sqrt{x}$ avec $[a, b] = [0, 1]$, x_1, \dots, x_n équirépartis sur $[a, b]$

$$\max_{a \leq t \leq b} |f(t) - P(t)| \xrightarrow{n \rightarrow \infty} +\infty$$

ce phénomène est appelé phénomène de Runge.

Il en reste une solution:

- si f est lipschitziennne sur $[a, b]$ ou Hölderienne

$$\exists a \in]0, 1[, |f(x) - f(y)| \leq C|x - y|$$

- Si x_1, \dots, x_n sont les racines du n-ème polynôme de Tchebychev.

$$|f(x) - P(x)| \leq \frac{|f^{(n)}(x)|}{n!} \prod_{i=1}^n (x - x_i)$$

2.5.2 Polynôme de Tchebychev

Definition 2.33. Les polynômes de Tchebychev sont définis par la recurrence:

$$\begin{cases} T_0 = 1 \\ T_1 = x \\ T_n = 2xT_{n-1} - T_{n-2} & n \geq 2 \end{cases}$$

Proposition 2.34. Le n-ième polynôme de Tchebychev vérifie:

1. T_n est de degré exactement n et son coefficient de plus haut degré est 2^{n-1} , $n \geq 1$
2. T_n a n racines distinctes simples

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

3. $|T_n(x)| \leq 1, \quad \forall x \in [-1, 1], |T_n(x)| = 1 \Leftrightarrow x \in \{x_0, \dots, x_n\} x_k = \cos(k\frac{\pi}{n})$

$$|T_n(x)| = 1 \text{ si } x \in \{x_k\} \quad x_k = \cos(k\frac{\pi}{n}) \quad (0 \leq k \leq n)$$

Proof. 1. Par récurrence:

Soit (P_n) la propriété " T_n est de degré n et son coef. de plus haut degré est 2^{n-1} ", $n \geq 1$. P_0 et P_1 vraies ($k \leq n$).

Supposons P_k vrai et montrons que P_{k+1} vrai.

En effet, nous avons $T_{n+1} = 2xT_n - T_{n-1}$, on en déduit que P_{n+1} est vraie.

Maintenant,

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

En effet, pour $\begin{cases} n = 0, T_0(x) = 1 = \cos(0) \\ n = 1, T_n(x) = \cos(\arccos(x)) \end{cases}$ et $n > 1$

$$\cos((n+1)\arccos(x)) = \cos(n\arccos(x))\cos(\arccos(x)) - \sin(n\arccos(x))\sin(\arccos(x))$$

$$\cos((n-1)\arccos(x)) = \cos(n\arccos(x))\cos(\arccos(x)) + \sin(n\arccos(x))\sin(\arccos(x))$$

On a:

$$\cos((n+1)\arccos(x)) = 2x\cos(n\arccos(x)) - \cos((n-1)\arccos(x))$$

D'où $x \mapsto \cos(n\arccos(x))$ vérifie la même récurrence sur $[-1, 1]$ que T_n . Par conséquent les 2 coïncident sur $[-1, 1]$. On en déduit $\forall x \in [-1, 1]$

2.

$$\begin{aligned} T_n(x) = 0 &\Leftrightarrow \cos(n\arccos(x)) = 0 \\ &\Leftrightarrow n\arccos(x) = \frac{\pi}{2} \pmod{\pi} \\ &\Leftrightarrow \arccos(x) = \frac{\pi}{2n} \pmod{\frac{\pi}{n}} \quad \Leftrightarrow x = \cos\left(\frac{\pi}{2n} + k\frac{\pi}{n}\right) \quad 0 \leq k \leq n-1 \end{aligned}$$

3. $|\cos(x)| \leq 1$ D'où $|T_n(x)| \leq 1, \forall x \in [-1, 1]$

$$\begin{aligned} |T_n(x)| = 1 &\Leftrightarrow n\arccos(x) = 0 \pmod{\pi} \\ &\arccos(x) = 0 \pmod{\frac{\pi}{n}} \end{aligned}$$

$$\in x \in \{\cos(k\frac{\pi}{n}), k \in [0, n]\}$$

□

Proposition 2.35. Si Q_n est un polyôme de degré n de même coeff. de plus haut degré que T_n , alors:

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

Corollary 2.36. Si ξ_1, \dots, ξ_n sont n points 2 à 2 distincts de $[-1, 1]$, on a:

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

où x_j sont les racines de T_n

2.5.3 Application

Soit ξ_1, \dots, ξ_n , 2 à 2 distincts, P le polynôme de lagrange de f (suffisament régulière), alors:

$$\begin{aligned} |f(x) - P(x)| &\leq \frac{\|f^{(n)}\|_\infty}{n!} |\omega_n(x)| \\ &\leq \frac{\|f^{(n)}\|_\infty}{n!} \|\omega_n(x)\|_\infty \end{aligned}$$

où $\omega_i = \prod_{j=1}^n (x - \xi_j)$ et $\|\cdot\|_\infty$ et loi norme inf sur $[-1, 1]$. Ainsi, le choix de ξ_i qui possede la plus petite valeur de $\|\omega_n\|_\infty$ est celui des racines du n-ième polynôme de Tchebychev.

Remark 2.37. On se ramène à un intervalle quelconque $[a, b]$ par

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

sont les racines des polynômes de Tchebychev sur $[a, b]$

CHAPTER 3

INTÉGRATION NUMÉRIQUE

But: On souhaite calculer au mieux

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

où $f : [a, b] \rightarrow \mathbb{R}$ donné

Contraintes

- f n'a pas de primitive connue (ou évidente)
- f n'est connue ou ne peut être évaluée qu'en un certain nombre fini de points

$$(x_i, 0 \leq i \leq n \text{ sur } [a, b])$$

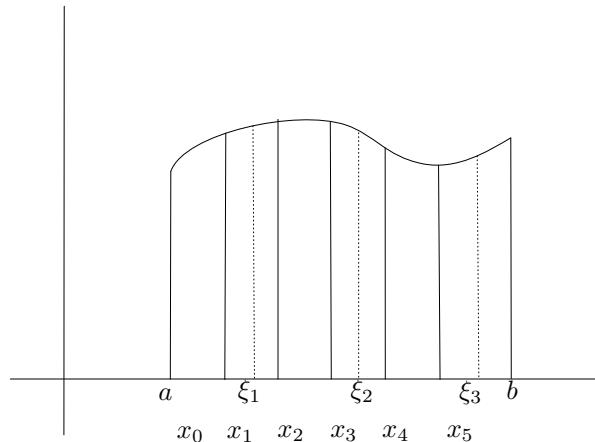


Figure 3.1: Exemple d'une intégration

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

$$S(f, \sum_N) = \sum_{i=0}^N f(\xi_i) \underbrace{(x_{i+1} - x_i)}_{\omega_i}$$

somme de Riemann associée à \sum_N . Théorème: $\lim_{N \rightarrow \infty} S(f, \sum_N) = \int_a^b f(x) dx$

3.1 Formule de quadrature

Definition 3.1. Étant donnée N points x_1, \dots, x_N de l'intervalle $[a, b]$ et N poids $\omega_1, \dots, \omega_N \in \mathbb{R}$ associées à chaque points.

On appelle formule de quadrature associé aux $(x_i), (\omega_i)$ l'application linéaire sur $C^0([a, b])$

$$\tilde{I}(f) = \sum_{i=1}^N \omega_i f(x_i)$$

On dit que la formule de quadrature est d'ordre p si elle est exacte pour les polynômes de degré p_1 . i.e

$$\tilde{I}(Q) = \int_a^b Q dx \quad \forall Q \in \mathbb{R}_{p-1}[X]$$

et s'il existe $Q \in \mathbb{R}_{p-1}[X]$ tq $I(Q) \neq \int_a^b Q dx$, autrement dit si elle exacte pour le polyôme de degré au plus $p - 1$.

Remark 3.2. On note:

$$\int_a^b f(x) dx \approx \sum_{i=1}^n \omega_i f(x_i)$$

| | | | |
|--------|------------|------------|-----|
| Points | x_1 | x_2 | ... |
| Poids | ω_1 | ω_2 | ... |

Example 3.3. Soit la formulle de quadrature

$$\int_a^b f(x) dx \approx (b-a)f(a)$$

- si $f = 1$, on a

$$\int_a^b f(x) dx = b-a = (b-a)f(a)$$

elle exacte pour les polynômes de degré 0.

- si $f(x) = x$ on a

$$\int_a^b f(x) dx = \left[\frac{x^2}{2} \right]_a^b = \frac{(b-a)(a+b)}{2} \neq (b-a)a$$

elle n'est pas exacte pour les polynômes de degré 1.

Conclusion: elle est exacte pour les polynômes de degré au plus 0. Elle est donné d'ordre 1.

3.1.1 Construction de formule de quadrature à points donnés

Proposition 3.4. Soit x_1, \dots, x_N , N points 2 à 2 distincts de $[a, b]$.

1. Il existe un unique $(\omega_1, \dots, \omega_N) \in \mathbb{R}^N$ tels que

$$\tilde{I}(Q) \stackrel{\text{def}}{=} \sum_{i=1}^N \omega_i Q(x_i) = \int_a^b Q(x) dx \quad \forall Q \in \mathbb{R}_{N-1}[X]$$

2. Pour toute fonction $f : [a, b] \rightarrow \mathbb{R}$ de classe \mathcal{C}^N

$$\left| \int_a^b f(x) dx - \tilde{I}(f) \right| \leq \frac{(b-a)^{N+1}}{N!} \|f^{(N)}\|_{\infty}$$

Proof. Soit $l_i, i = 1, \dots, N$ la base de Lagrange associé aux x_i , i.e

$$l_i(x) = \prod_{j=1, j \neq i}^N \frac{X - x_j}{x_i - x_j}$$

on a $l_i \in \mathbb{R}_{N-1}[X]$.

Soit $Q \in \mathbb{R}_{N-1}[X]$, Q coïncide avec le polynôme d'interpolation de lagrange aux points x_1, \dots, x_N

$$Q(X) = \sum_{i=1}^N Q(x_i) l_i(X)$$

d'où

$$\begin{aligned} \int_a^b Q(x) dx &= \sum_{i=1}^N Q(x_i) \int_a^b l_i(x) dx \\ &= \sum_{i=1}^n Q(x_i) \omega_i \end{aligned}$$

où $\omega_i = \int_a^b l_i(x) dx$. D'où l'existence.

Unicité: Soit $\tilde{\omega}_i, i = 1, \dots, N$,

$$k : \int_a^b Q(x) dx = \sum_{i=1}^N \tilde{\omega}_i Q(x_i) \quad \forall Q \in \mathbb{R}_{N-1}[X]$$

alors, puisque $l_i \in \mathbb{R}_{N-1}[X]$, on a

$$\int_a^b l_i(x) dx = \tilde{\omega}_i \quad i = 1, \dots, N$$

D'où ($\tilde{\omega}_i = \omega_i$) et on a l'unicité

Estimation d'erreur:

Soit f de classe \mathcal{C}^N sur $[a, b]$ et R_f un poly d'interpolation aux points $x_i \quad i = 1, \dots, N$. On a:

$$\begin{aligned} \tilde{I}(f) &= \sum_{i=1}^N f(x_i) \omega_i = \sum_{i=1}^N P_f(x_i) \omega_i \\ &= \int_a^b P_f(x) dx \end{aligned}$$

D'où

$$\begin{aligned} \left| \int_a^b f(x) dx - \tilde{I}(f) \right| &= \left| \int_a^b f(x) dx - \int_a^b P_f(x) dx \right| \\ &\leq \int_a^b |f(x) - P_f(x)| dx \\ &\leq \frac{\|f^{(N)}\|_{\infty} (b-a)^N}{N!} (b-a) \end{aligned}$$

$$\left| \int_a^b f(x) dx - \tilde{I}(f) \right| \leq \frac{\|f^{(N)}\|_\infty}{N!} (b-a)^N (b-a)$$

□

python:

```
1 from scipy.integrate import quad
2 quad(f, a, b) =
```

3.2 Utilisation des formules de quadrature. Formule de quadrature élémentaires et composées

3.2.1 Définition

On appelle formule de quadrature élémentaire I_e sur $[-1, 1]$ associée aux poins $\xi_i \in [-1, 1]$ et de poids ω_i , $i = 1, \dots, N$, la formule

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

Une subdivision d'un intervalle $[a, b]$ en n points $a = x_1 < x_2 < \dots < x_n = b$ étant donnée, elle induit une formule de quadrature composite I_c pour $[a, b]$

$$I_c = \sum_{i=1}^{n-1} I_e(f; x_i, x_{i+1}) \equiv \sum_{j=1}^{n-1} \frac{x_{j+1} - x_j}{2} \sum_{i=1}^N \omega_i f(x_{i,j}) \text{ où } x_{i,j} = \frac{(x_j + x_{j+1}) + \xi_i(x_{j+1} - x_j)}{2}$$

Illustration:

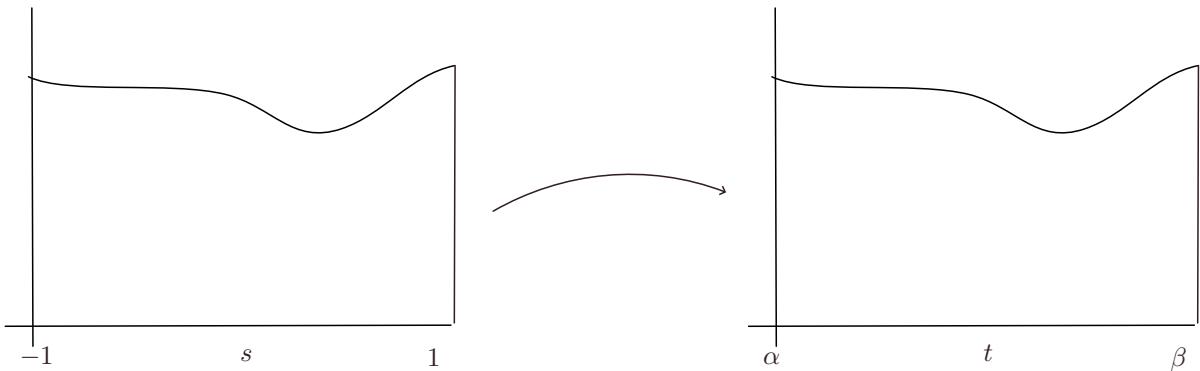


Figure 3.2: $I_e(f) = \sum_{i=1}^N \omega_i f(\xi_i) \approx \int_\alpha^\beta f(t) dt$

$$\begin{aligned} \int_\alpha^\beta f(t) dt &\approx \frac{s+1}{2} = \frac{t-\alpha}{\beta-\alpha} \Rightarrow \eta = \alpha + \frac{s+1}{2}(\beta-\alpha) \\ dt &= \frac{\beta-\alpha}{2} ds \end{aligned}$$

$$\begin{aligned}
\int_{\alpha}^{\beta} f(t) dt &= \frac{\beta - \alpha}{2} \int_{-1}^1 f(\alpha + \frac{s+1}{2}(\beta - 1)) ds \approx \frac{\beta - \alpha}{2} \sum_{i=1}^N \omega_i f(\underbrace{\alpha + \frac{\xi_i + 1}{2}(\beta - 1)}_{x_i}) \\
&\approx \sum_{i=1}^N \left(\frac{\beta - \alpha}{2} \omega_i \right) f(x_i) \\
&\approx I_e(f, \alpha, \beta)
\end{aligned}$$

3.2.2 Méthode de quadrature clamique

Méthode de rectangles

Definition 3.5.

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

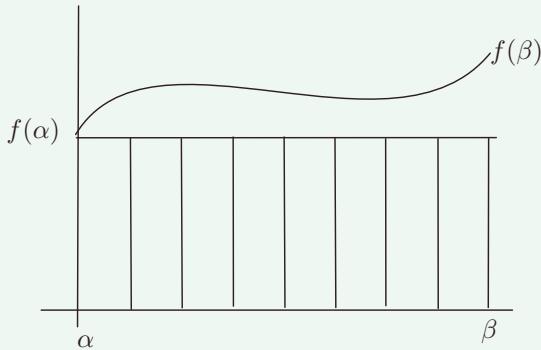


Figure 3.3: methode-rectangle

Proposition 3.6. • Si f est \mathcal{C}^1 sur $[\alpha, \beta]$ l'erreur de la méthode de quadrature élémentaires :

$$E_e(f) = I(f) - I_e(f) = \frac{f'(c)}{2}(\beta - \alpha)^2 \text{ où } c \in]\alpha, \beta[$$

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

$$|E_c(f)| = |I(f) - I_c(f)| \leq h \underbrace{\|f'\|_{\mathcal{C}^0([a,b])}}_{\max_{a \leq x \leq b} |f'(x)|} \frac{b-a}{2}$$

Proof. Posons $F(x) = \int_a^x f(t) dt$, F est de \mathcal{C}^2 car f est \mathcal{C}^1 . Par Taylor lagrange à l'ordre 2 en α donne

$$F(\beta) = F(\alpha) + F'(\alpha)(\beta - \alpha) + F''(c) \frac{(\beta - \alpha)^2}{2} \text{ où } c \in]\alpha, \beta[$$

D'où

$$\int_{\alpha}^{\beta} f(t) dt - f(\alpha)(\beta - \alpha) = \frac{f'(c)}{2}(\beta - \alpha)^2$$

On en déduit

$$\begin{aligned}|I(f) - I_c(f)| &= \left| \int_a^b f(t) dt - I_c(f) \right| \\&= \left| \sum_{j=0}^{n-1} \int_{x_j}^{x_{j+1}} f(t) dt - \sum_{j=0}^{n-1} f(x_j)(x_{j+1} - x_j) \right| \\&\leq \sum_{j=0}^{n-1} \left| \int_{x_j}^{x_{j+1}} f(t) dt - f(x_j)(x_{j+1} - x_j) \right| \\&\leq \sum_{j=0}^{n-1} \frac{\|f'\|_{\infty}}{2} (x_{j+1} - x_j)^2 \\&\leq h \frac{\|f'\|_{\infty}}{2} \sum_{j=0}^{n-1} (x_{j+1} - x_j) \\&\leq h \frac{\|f'\|_{\infty}}{2} (b - a)\end{aligned}$$

□

3.2.3 Méthode de trapèze

Definition 3.7.

$$\int_a^b f(x) dx \approx \frac{f(a)f(b)}{2}(b - a)$$

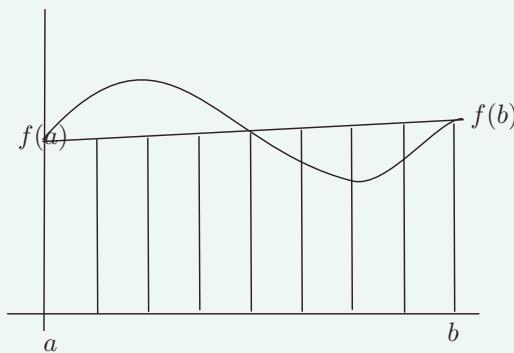


Figure 3.4: Méthode de trapèze

Proposition 3.8. Si f est \mathcal{C}^2 sur $[a, b]$

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

Proof. La formule était exacte pour les poly de degré 1, elle est du type interpolation:

$$I_c(f) = \int_a^b P_f(t) dt \text{ où } P_f \text{ est le poly d'ext aux points } a, b$$

D'où

$$\begin{aligned} |E_c(f)| &= |I(f) - I_c(f)| = \left| \int_a^b (f(t) - P_f(t)) dt \right| \\ &\leq \frac{\|f''\|_\infty}{2!} \int_a^b (x-a)(b-x) dx \end{aligned}$$

On en déduit que:

$$\begin{aligned} |I(f) - I_c(f)| &= \left| \sum_{j=0}^{n-1} \left(\int_{x_j}^{x_{j+1}} f(t) dt \right) - \frac{f(x_j) + f(x_{j+1})}{2} (x_{j+1} - x_j) \right| \\ &\leq \sum_{j=0}^{n-1} |E_c(f)| \\ &\leq \frac{\|f''\|_\infty}{12} \sum_{j=0}^{n-1} h^3 = \frac{h^2}{12} \|f''\|_\infty \sum_{j=0}^{n-1} h \\ &\leq \frac{1}{12} h^2 \|f''\|_\infty (b-a) \end{aligned}$$

□

3.2.4 Méthode de point-milieu

Definition 3.9.

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

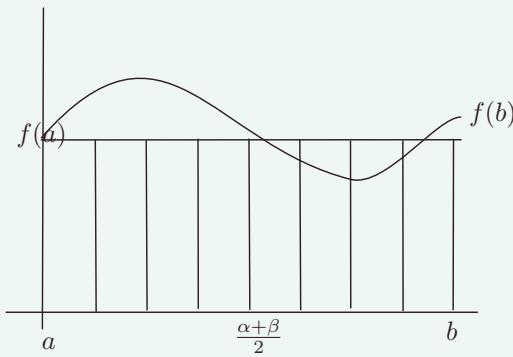


Figure 3.5: methode-de-point-milieu

Example 3.10. degré d'exactitude sur $[-1, 1]$ elle devient

$$\int_{-1}^1 f(s) ds \approx 2f(0)$$

- Pour $f = 1$, on a $I(f) = \int_{-1}^1 ds = 2 = I_c(f)$. D'où I_c est exacte pour poly de degré au moins 0.
- Pour $f = s$ on a $I(f) = \int_{-1}^1 s ds = 0 = 2 \cdot 0 = I_c(f)$. D'où I_c est exacte pour poly de degré au moins 1.
- Pour $f = s^2$ on a $I(f) = \int_{-1}^1 s^2 ds = \frac{2}{3} \neq 2 \cdot 0^2 = I_c(f)$. La formule élémentaire n'est pas exacte pour poly de degré 2. Donc, la formule de point-milieu est exacte pour les polynômes de degré 1.

Proposition 3.11. • Si f est $\mathcal{C}^2([a, b])$ on a:

$$E_c(f) = I(f) - I_c(f) = \frac{f''(c)}{24}(b-a)^3$$

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

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

Proof. exo □

3.2.5 Méthode de Simpson

Definition 3.12.

$$\int_{\alpha}^{\beta} f(t) dt \approx \frac{\beta - \alpha}{6} \left(f(\alpha) + 4f\left(\frac{\alpha + \beta}{2}\right) + f(\beta) \right)$$

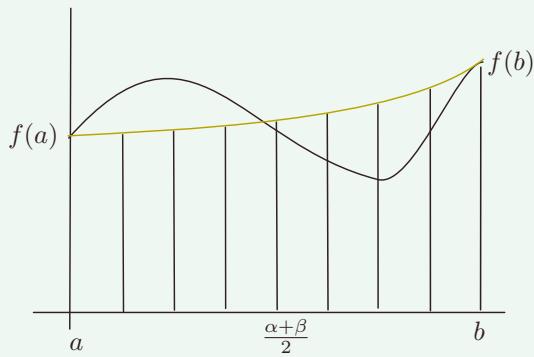


Figure 3.6: methode-de-simpson

Proposition 3.13. • Si f est \mathcal{C}^{3+1} sur $[\alpha, \beta]$ on a:

$$E_c(f) = \int_{\alpha}^{\beta} f(x) dx - I_e(f) = -\frac{f^{(4)}}{2880}(\beta - \alpha)^5 \text{ où } c \in]\alpha, \beta[$$

$$|E_c(f)| \leq h^4 \|f^{(4)}\|_{\infty} \frac{(b-a)}{2880}$$

3.2.6 Méthode de Newton-Côte

C'est une généralisation des méthodes élémentaires.

Definition 3.14. On appelle méthode de Newton-Côte d'ordre k la méthode élémentaire construite en utilisant le polynôme d'intégration d'ordre k , associé aux $k + 1$ points équidistants

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

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

où $x_i = \alpha + i \frac{\beta - \rho}{k}$, $i = 0, \dots, k$ et

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

Remark 3.15. • Cette formule est $\begin{cases} \text{d'ordre } k \text{ si } k \text{ impair} \\ \text{d'ordre } k + 1 \text{ si } k \text{ pair} \end{cases}$

- On n'utilise les méthodes que pour k pair sauf le cas $k = 1$
- Si $k = 1$ on a la formule des trapèzes

- Si $k = 2$ on a la formule de Simpson
- Si $k = 4$ on a la formule de Boole-Villarceau

$$\int_{-1}^1 f(x) dx \approx \frac{7}{90}f(-1) + \frac{16}{49}f\left(-\frac{1}{2}\right) + \frac{2}{15}f(0) + \frac{16}{45}f\left(\frac{1}{2}\right) + \frac{7}{90}f(1)$$

- Pour $k = 6$, on a la formule de Hordy
- Pour $k \geq 8$ on a des points ω_i , négatifs, ce qui rendent les formules semblables aux erreurs d'arrondie.

Theorem 3.16. Soient $I(f) = \int_{\alpha}^{\beta} f(x) dx$, $I_e(f) = \sum_{i=0}^k \omega_i f(x_i)$, $E(f) = I(f) - I_e(f)$. Supposons que la méthode d'intégration soit d'ordre $p \geq k$. Posons

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

avec $x_+ = \begin{cases} x & \text{si } x \geq 0 \\ 0 & \text{sinon} \end{cases}$. On a:

$$E(t) = \int_{\alpha}^{\beta} \frac{K(t)}{p!} f^{(p+1)}(t) dt \quad \forall f \in \mathcal{C}^{p+1}([\alpha, \beta])$$

Si de plus K est de signe constante sur $[\alpha, \beta]$, il existe $c \in [\alpha, \beta]$ telle que

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

On appelle Noyau de Peano associée à la méthode, la fonction

$$t \mapsto \frac{K(t)}{p!}$$

Proof. Formule de Taylor avec reste intégrale:

$$f(x) = \sum_{i=0}^l \frac{(x-\alpha)^i}{i!} + \int_{\alpha}^x \frac{(x-t)^p}{p!} f^{l+1}(t) dt$$

$$E(f) = E\left(\sum_{i=0}^l \frac{(x-\alpha)^i}{i!}\right) + E\left(\int_{\alpha}^x \frac{(x-t)^p}{p!} f^{l+1}(t) dt\right)$$

□

Remark 3.17. Lorsque K est de signe constante,

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

D'où

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

Dans les méthodes et Newton-Côte le noyau de Peano a une signe constante.

3.3 Construction de formule de quadrature (à points inconnus): Formules de Gauss Legendre

On cherche s'il existe un meilleur choix des points x_1, \dots, x_n de $[\alpha, \beta]$ tel que la formule de quadrature associée soit exacte sur $\mathbb{R}_{n'}[X]$ pour $n' > n$.

Example 3.18. Cherchons une table fomrulé à 2 ponts

$$\int_{-1}^1 f(x) dx \approx \omega_1 f(x_1) + \omega_2 f(x_2)$$

En effet, on a 4 inconnus, il faut donc 4 équations. On pose comme condition l'exactitude de cette formule pour les polynômes $1, x, x^2, x^3$, d'où

$$\begin{cases} 2 = \omega_1 + \omega_2 \\ 0 = \omega_1 x_1 + \omega_2 x_2 \\ \frac{2}{3} = \omega_1 x_1^3 + \omega_2 x_2^3 \\ 0 = \omega_1 x_1^3 + \omega_2 x_2^3 \end{cases}$$

Remark 3.19. x_1 et x_2 sont racines du polynôme $\frac{1}{2}(3x^2 - 1)$ i.e $x_1 = -\frac{1}{\sqrt{3}}$ $x_2 = \frac{1}{\sqrt{3}}$. D'où $\omega_1 = \omega_2 = 1$.

D'où

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

est exacte sur $\mathbb{R}_3[X]$ ($3 = 2 \cdot 2 - 1$)

Example 3.20. Formule à 3 points

$$\int_{-1}^1 f(x) dx \approx \omega_1 f(x_1) + \omega_2 f(x_2) + \omega_3 f(x_3)$$

On impose l'exactitude $1, x, x^2, x^3, x^4, x^5$

$$\begin{aligned} \left| \int_{\alpha}^{\beta} f(x) dx - I_e[f, \alpha, \beta] \right| &\leq c(\beta - \alpha)^{5+2} \\ \left| \int_{\alpha}^{\beta} f(x) dx - I_c[f, \alpha, \beta] \right| &\leq ch^6 \end{aligned}$$

On obtient x_i sont racines du polynôme $\frac{1}{3}(5x^3 - 3x)$. D'où, $x_1 = -\sqrt{\frac{3}{5}}$, $x_2 = 0$, $x_3 = \sqrt{\frac{3}{5}}$ et $\omega_1 = \frac{5}{9}$, $\omega_2 = \frac{8}{9}$, $\omega_3 = \frac{5}{9}$. D'où

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

Proposition 3.21. Considérons la formule à n points

$$\int_{-1}^1 f(x) dx \approx \omega_1 f(x_1) + \dots + \omega_n f(x_n)$$

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 définie par la recurrence.

$$\begin{aligned}L_0(x) &= 1 \\L_1(x) &= x \\L_{n+1}(x) &= \frac{1}{n+1} [(2n+1)xL_n(x) - nL_{n-1}(x)]\end{aligned}$$

$$\omega_i = \int_{-1}^1 \prod_{\substack{j=1 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j} dx \quad i = 1, 2, \dots, n$$

La formule de quadrature ainsi construite est appellé formule de Gauss-Legendre.

$$\int_{-1}^1 f(x) dx \approx \omega_1 f(x_1) + \omega_2 f(x_2)$$

exacte pour $1, x, x^2, x^3$ pour $(x - x_1)(x - x_2)$

$$\int_{-1}^1 (x - x_1)(x - x_2) dx = 0$$

$$\int_{-1}^1 x(x - x_1)(x - x_2) dx = 0$$

$$\int_{-1}^1 x^2 - (x_1 + x_2)x + x_1 x_2 dx = 0$$

$$\begin{cases} \frac{2}{3} + (x_1 x_2)2 = 0 \\ -\frac{2}{3}(x_1 x_2) = 0 \end{cases}$$

$$\int_{-1}^1 f(x) dx \approx \omega_1 f(-\frac{1}{\sqrt{3}}) + \omega_2 f(\frac{1}{\sqrt{3}})$$

- $f \equiv (x - \frac{1}{\sqrt{3}}) \equiv (x - x_1) \Rightarrow -x_1 2 = \omega_1 2 x_1$

- $f \equiv (x + \frac{1}{\sqrt{3}}) \equiv (x - x_2)$

CHAPTER 4

RÉSOLUTION APPROCHÉE D'ÉQUATIONS DIFFÉRENTIELLES ORDINAIRES (EDOs)

4.1 Motivations

4.1.1 Définitions

Definition 4.1. Soit

$$\begin{aligned} f : [a, b] \times \mathbb{R}^d &\longrightarrow \mathbb{R}^d \\ (t, x) &\longmapsto f((t, x)) \end{aligned}$$

avec $a, b \in \mathbb{R}$ et $d \in \mathbb{N}^*$ donnée par des d composantes

$$\begin{aligned} f_i : [a, b] \times \mathbb{R}^d &\longrightarrow \mathbb{R}^d \\ (t, x) &\longmapsto f_i(t, x) \end{aligned}$$

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

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

$$(g \in \mathcal{C}^1([a, b], \mathbb{R}), i = 1, \dots, d) \Leftrightarrow (g \in \mathcal{C}^1([a, b], \mathbb{R}^d))$$

Definition 4.2. On appelle équation différentielle d'ordre 1 une équation de la forme

$$y'(t) = f(t, y(t)), \quad t \in]t_0, t_0 + \tau[$$

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 continue.

Definition 4.3. -

- Une fonction y de classe \mathcal{C}^1 vérifiant une EDO est dite solution de l'EDO.
- Résoudre une EDO c'est déterminer toutes les solutions de cette EDO.
- Lorsque $d \neq 1$, on parle de système d'EDOs.

Remark 4.4. Toute EDO d'ordre $p > 1$ peut être ramenée à un système d'EDOs d'ordre 1.

Definition 4.5. On appelle condition de Cauchy de l'EDO, la donnée de la valeur de la solution en un point

$$t_0 \in [a, b] : \quad y(t_0) = y^0$$

Le couple (t_0, y^0) est appelé **condition initiale** et le problème de Cauchy consiste à la recherche d'une fonction de classe \mathcal{C}^1 vérifiant:

$$\begin{cases} y'(t) = f(t, y(t)) & t \in]t_0, t_0 + \tau[\\ y(t_0) = y^0, & t_0 \text{ donné des } \mathbb{R}^d \end{cases}$$

4.1.2 Exemple

Example 4.6. Pendule

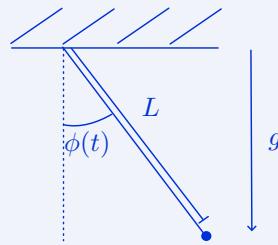


Figure 4.1: pendule-exemple-edo

$$\begin{cases} m = 1 \\ \phi(t) = ? \\ \phi' + \frac{g}{L} \sin(\phi) = 0 \end{cases}$$

c'est une EDO d'ordre 2.

$$\begin{cases} x_1 = \phi \Rightarrow x'_1 = \phi' = x_2 \\ x_2 = \phi' \Rightarrow x'_2 = \phi'' = -\frac{g}{L} \sin(\phi) = -\frac{g}{L} \sin(x_1) \end{cases}$$

D'où $X(t) = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ on a $X'(t) = \begin{pmatrix} x_2 \\ -\frac{g}{L} \sin(x_1) \end{pmatrix} = f(t, X(t))$ où

$$\begin{aligned} f : \mathbb{R} \times \mathbb{R}^2 &\longrightarrow \mathbb{R}^2 \\ (t, X \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}) &\longmapsto f((t, X \begin{pmatrix} x_1 \\ x_2 \end{pmatrix})) = \begin{pmatrix} x_2 \\ -\frac{g}{L} \sin(x_1) \end{pmatrix}. \end{aligned}$$

Example 4.7. Objet en chasse libre.



Figure 4.2: asteroid-edo

$$\begin{cases} \text{vitesse: } v \\ \text{altituded: } z \end{cases}$$

k a coef. de frottement.

$$z'' = -g + k(z')^2 e^{-az} \text{ càd EDO d'ordre 2}$$

ou encore

$$\begin{cases} z' = v \\ v' = -g + kv^2 e^{-az} \end{cases} \text{ càd système d'EDOs d'ordre 1}$$

$$\text{Posons } Y = \begin{pmatrix} z \\ z' \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \quad Y = f(t, Y) ???$$

$$f(t, Y \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}) = \begin{pmatrix} y_2 \\ -g + ky_2^2 e^{-ay_1} \end{pmatrix}$$

```

1 def f(t, Y):
2     x, y = Y
3     return np.array([y, -g + (k*y**2)*np.exp(-a*x)])

```

Example 4.8. Taux d'inféction

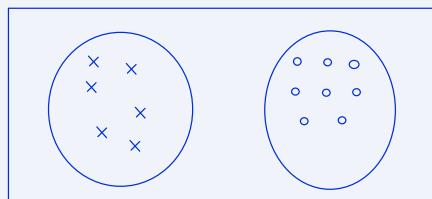


Figure 4.3: taux-d-infection-edo

y : infectés, x : soins, α : taux d'inféction

$$\begin{cases} \dot{y} = \alpha xy \\ x + y = n \end{cases} \Rightarrow y' = \alpha y(n - y)$$

4.1.3 Nécessite de la solution approchée

On considère le problème de Cauchy

$$\begin{cases} x'(t) = f(t, x(t)) & t \in]t_0, t_0 + \tau[\\ x(t_0) = x^0 \in \mathbb{R}^d \end{cases} \quad (4.1)$$

On ne sait résoudre 4.1 dans des cas particuliers.

$d = 1$, f est à variables séparées.

Example 4.9.

$$\begin{cases} \dot{L} = \tau_L L &]0, T[\\ L(0) = L_0 \end{cases} \Rightarrow L(t) = L_0 e^{t\tau_L}$$

Illustration graphique

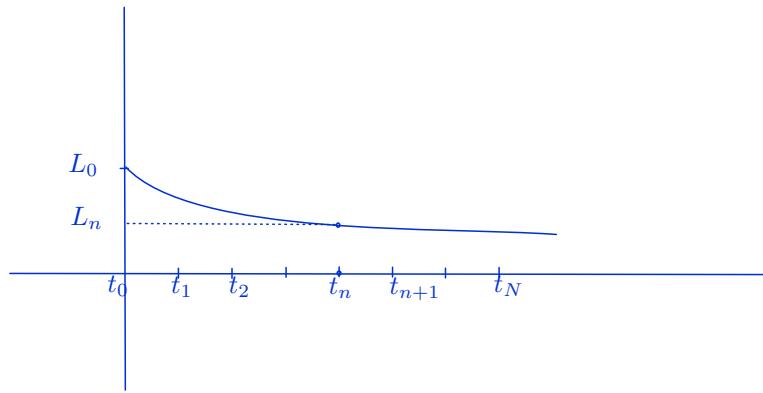


Figure 4.4: illustration-graphique-sol-approche

- On se donne $t_n = n\Delta t$

$$n = 0, \dots, N, \text{ où } \Delta t = \frac{T}{N}, N \in \mathbb{N}^*$$

- On calcule $L_n = L_0 e^{t_n \tau_L}, n = 0, \dots, N$
- On place (t_n, L_n) sur un figure et on les relie pour obtenir un graphe de $t \mapsto L(t)$

4.2 Problème d'évolution de population des lapins

L : population des lapins, R : population renards.

On a le problème de Cauchy

$$\begin{cases} \dot{L} = L(\tau_L - pR) \\ \dot{R} = R\tau_R(\alpha L - 1) \\ L(0) = L_0, \quad R(0) = R_0 \end{cases}$$

Ce système n'est pas résoluble analytiquement.

On peut cependant le résoudre numériquement à condition de s'assurer que le problème est bien posé.

- Existence et l'unicité de la solution
- Régularité de la solution
- Dépendance continue de la solution vis à vis des données du problème (où Stabilité)

$$\begin{aligned} \Phi : (t_0, f) &\mapsto y \\ \|\Phi(y_1 - y_2)\|_* &\leq C_1 \|L - L_0\|^2 + C_2 \|f_1 - f_2\|_{**} \end{aligned}$$

Definition 4.10. On dit que $f : [a, b] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ est lipschizienne par rapport à sa seconde variable s'il existe une constante positive L (appelée constante de lipschitz) telle que

$$\|f(t, y_2) - f(t, y_1)\| \leq L\|y_2 - y_1\| \quad \forall t \in [a, b] \forall y_1, y_2 \in \mathbb{R}^d$$

Theorem 4.11. de Cauchy lipshitz.

Soit

$$\begin{cases} x'(t) = f(t, x(t)), & t \in]t_0, t_0 + \tau[\\ x(t_0) = x^0 \end{cases} \quad (4.2)$$

- Si $f : [t_0, t_0 + \tau] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ vérifie

1. f continue
2. $\|f(t, y) - f(t, z)\| \ll \|y - z\|$

Alors 4.2 admet une unique solution (globale) de classe $C^1([t_0, t_0 + \tau], \mathbb{R}^d)$

4.3 Exemple de Schémas numériques

4.3.1 Formulation intégrale

Proposition 4.12. x solution de 4.2 ssi $x(t) = x^0 + \int_{t_0}^t f(s, x(s)) ds \forall t \in [t_0, t_0 + \tau]$

Proof. -

$\Rightarrow)$

$$\begin{cases} x(t_0) = x^0 \\ x'(t) = f(t, x(t)) \end{cases}$$

$\Leftarrow)$

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

□

4.3.2 Construction de schema d'Euler explicite

Étape 1 maillage du domaine

$$N \text{ donné, pose } \Delta t = \frac{T}{N}, t_n = n\Delta t, n = 0, \dots, N$$

Étape 2 : Formulation intégrale:

Suite $[t_n, t_{n+1}]$ problème discrèt ("continue")

$$\begin{cases} x(t_{n+1}) = x(t_n) + \int_{t_n}^{t_{n+1}} f(s, x(s)) ds, n = 0, \dots, N-1 \\ x(t_0) = x^0 \end{cases} \quad (4.3)$$

Étape 3 Approximation des intégrales (Formules de quadratures)

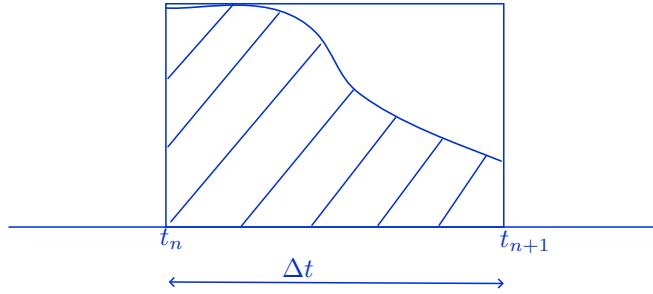


Figure 4.5: etape-3-rectangles-a-gauche

Récrangles à gauche

$$\int_{t_n}^{t_{n+1}} g(s) ds \Delta t g(t_n) + o(\Delta t^2) \approx \Delta t g(t_n)$$

On a

$$\begin{cases} x(t_{n+1}) = x(t_n) + \Delta t f(t_n, x(t_n)) + o(\Delta t^2) \\ x(t_0) = x^0 \end{cases} \quad (4.4)$$

$$\begin{cases} x(t_{n+1}) \approx x(t_n) + \Delta t f(t_n, x(t_n)) \\ x(t_0) = x^0 \end{cases} \quad (4.5)$$

On pose $x_n \approx x(t_n)$, $n = 0, \dots, N$ lorsque dans (PDC) on se sépare des restes

$$\begin{cases} x_{n+1} = x_n + \Delta t f(t_n, x_n), & n = 0, \dots, N - 1 \\ x_0 = x^0 \end{cases} \quad (4.6)$$

On a le schéma d'Euler explicite

Remark 4.13. Schéma du point-millieu

$$\int_{t_n}^{t_{n+1}} g(s) ds \Delta t g\left(\frac{t_{n+1} + t_n}{2} + o(\Delta t^3)\right)$$

On aurait

$$x(t_{n+1}) = x(t_n) + \Delta t 3f(t_{n+\frac{1}{2}}, x(t_{n+\frac{1}{2}})) + o(\Delta t^3)$$

Soit

$$\begin{cases} x_{n+1} = x_n + \Delta t f(t_{n+\frac{1}{2}}, x_{n+\frac{1}{2}}), & n = 0, \dots, N - 1 \\ x_0 = x^0 \end{cases}$$

Comme $x_{n+\frac{1}{2}}$ est inconnu, on l'approche par le schéma d'Euler explicite.

$$\text{i.e. } x_{n+\frac{1}{2}} = x_n + \frac{\Delta t}{2} f(t_n, x_n)$$

D'où

$$\begin{cases} x_{n+\frac{1}{2}} = x_n + \frac{\Delta t}{2} f(t_n, x_n) \\ x_{n+1} = x_n + \Delta t f(t_n + \frac{\Delta t}{2}, x_{n+\frac{1}{2}}) \end{cases} \quad (4.7)$$

$$\begin{cases} x_{n+1} = x_n + \Delta t f(t_n + \frac{\Delta t}{2}, x_n + \frac{\Delta t}{2} f(t_n, x_n)), n = 0, \dots, N-1 \\ x_0 = x^0 \end{cases} \quad (4.8)$$

4.3.3 Autres schémas et forme général des schémas explicites à un pas

Remark 4.14. Si Rectangles à droits, on aurait eu

$$\begin{cases} x_{n+1} = x_n + \Delta t f(t_{n+1}, x_{n+1}), n = 0, \dots, N-1 \\ x_0 = x^0 \end{cases} \quad (4.9)$$

Il est implicite c'est le schéma d'Euler implicite

Remark 4.15. Formule des trapèze - on aurait

$$\begin{cases} x_{n+1} = x_n + \frac{\Delta t}{2} (f(t_n, x_n) + f(t_{n+1}, x_{n+1})), n = 0, \dots, N-1 \\ x_0 \text{ donné} \end{cases} \quad (4.10)$$

C'est le schéma de CLANK-NICOLAS il est implicite. On peut expliciter le schéma de C-N.

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

On aura

$$\begin{cases} x_{n+1} = x_n + \frac{\Delta t}{2} [f(t_n, x_n) + f(t_{n+1}, x_n + \Delta t f(t_n, x_n))], \\ x_0 \text{ donné} \end{cases} \quad (4.11)$$

$$\begin{cases} x_{n+1}^* = x_n + \Delta t f(t_n, x_n) & n = 0, \dots, N-1 \\ x_{n+1} = x_n + \frac{\Delta t}{2} [f(t_n, x_n) + f(t_{n+1}, x_{n+1}^*)] \end{cases} \quad (4.12)$$

C'est un schéma explicite appelé schéma de Heun. Généralisation des schéma à un pas.

$$\begin{cases} x_{n+1} = x_n + \Delta t \Phi(t_n, x_n, \Delta t), & n = 0, \dots, N-1 \\ x_0 \text{ donné} \end{cases} \quad (4.13)$$

où $\Phi : \mathbb{R} \times \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}^d$

Example 4.16. • Euler explicite : $\Phi(t, y, \Delta t) = f(t, y)$

- Point - Millieu : $\Phi(t, y, \Delta t) = f(t + \frac{\Delta t}{2}, y + \frac{\Delta t}{2} f(t, y))$
- HEUN: $\Phi(t, y, \Delta t) = \frac{1}{2} [f(t, y) + f(t + \Delta t, y + \Delta t f(t, y))]$

$$\begin{cases} x_{n+1} = x_n + \Delta t f(t_n + \frac{\Delta t}{2}, x_n + \frac{\Delta t}{2} f(t_n, x_n)) \\ x_0 = x^0 \end{cases} \quad (4.14)$$

4.4 Étude de schémas pokes EDOs

On aimerait savoir si la suite $(x_n)_n$ générée par S converge. Si oui, "vers" la solution de P .

4.4.1 Définition

Definition 4.17. le schéma S est dit convergent ssi pour toute solution x de P et toute suite de $(x_n)_n$ construite par S courbe $x^0 = x(t_0)$ on a

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

Où $x(t_n) - x_n =: e_n$ erreur globale à l'itération n

- S'il existe $C > 0$, ne dépendant que de f, t_0, T tq

$$\max_{0 \leq n \leq N} \|e_n\| \leq C \Delta t^p \quad \forall t \in [0, \Delta t_0]$$

Le schéma est dit convergent d'ordre p .

Remark 4.18. Pour étudier un schéma pour EDO, on procède en 2 étapes:

- On cherche l'ordre de consistance
- Puis, on regarde sa stabilité

4.4.2 Ordre d'un schéma à un pas explicite

Definition 4.19. (erreur de consistance)

Soit x la solution de P . On appelle erreur locale de troncature du schéma S à l'instant t_n la quantité

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

$$P \begin{cases} x'(t) = f(t, x(t)) & t \in]t_0, t_0 + T[\\ x^0 = x(t_0) \end{cases}$$

Definition 4.20. S est dit consistant d'ordre au moins q si pour toute solution exacte x de P , il existe une constante $C > 0$ telle que

$$\xi(t, \Delta t) = x(t + \Delta t) - x(t) - \Delta t + \Phi(t, x(t), \Delta t)$$

vérifie

$$\|\xi(t, \Delta t)\| \leq C \Delta t^{q+1}$$

- Si S est d'ordre au moins q mais pas d'ordre au moins $q + 1$, alors il est dit consistant d'ordre exactement q .

Example 4.21. Étude de consistance du Schéma d'Euler explicite

$$S \begin{cases} x_{n+1} = x_n + \Delta f(t_n, x_n), & n = 0, \dots, N \\ x^0 = x^0 \end{cases}$$

- L'erreur de consistance à l'instant t est

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

où x est solution suffisamment régulière de P (i.e $x'(t) = f(t, x(t))$)

- On cherche le petit o de Δt dans $\xi(t, \Delta t)$.

Comme x est solution exacte, on a:

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

Effectons les Δt

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

D'où

$$\xi(t, \Delta t) = \Delta t^2 \frac{x''(t)}{2} + o(\Delta t^2)$$

càd $\xi(t, \Delta t) = o(\Delta t^2)$ d'où le schéma est consistant d'ordre au moins 1. Or pour le problème $\begin{cases} x^0(t) = f(t, x(t)) \\ \text{avec } f(t, y) = 2t, \end{cases}$, on a la solution exacte $x(t) = x_0 + \frac{t^2}{2}$. Et on a:

$$\xi(t, \Delta t) = \Delta t^2$$

D'où le schéma ne peut pas être consistant d'ordre au moins 2. Il est donc consistant d'ordre exactement 1.

Proposition 4.22. Soit $\Phi : I \times \mathbb{R}^d \times [0, \Delta t_0] \rightarrow \mathbb{R}^d$ telle que $\frac{\partial \Phi}{\partial \Delta t}, \frac{\partial^2 \Phi}{\partial \Delta t^2}, \dots, \frac{\partial^q \Phi}{\partial \Delta t^q}$ existent et sont continues sur $I \times \mathbb{R}^d \times [0, \Delta t_0]$, f étant de classe C^q . Alors le schéma

$$S \begin{cases} x_{n+1} = x_n + \Delta t \Phi(t_n, x_n, \Delta t) & n = 0, \dots, N-1 \\ x_0 = x^0 \end{cases}$$

est consistant d'ordre au moins q ssi,

$$\begin{aligned} \Phi(t, y, 0) &= f(t, y) \\ \frac{\partial \Phi}{\partial \Delta t}(t, y, 0) &= \frac{1}{2} f^{[1]}(t, y) \\ &\vdots \\ \frac{\partial^{q-1} \Phi}{\partial \Delta t^{q-1}}(t, y, 0) &= \frac{1}{q} f^{[q-1]}(t, y) \end{aligned}$$

où $f^{[0]}(t, y) = f(t, y)$, $f^{[j]}(t, y) = \frac{\partial f^{[j-1]}}{\partial t}(t, y) + f(t, y) \frac{\partial f^{[j-1]}}{\partial y}(t, y)$ et $j = 1, \dots, q-1$

Example 4.23. Étude du schéma du PM

$$\begin{cases} x_{n+\frac{1}{2}} = x_n + \frac{\Delta t}{2} f(t_n, x_n) \\ x_{n+1} = x_n + \Delta t f(t_{n+\frac{1}{2}}, x_{n+\frac{1}{2}}) \\ x_0 = x^0 \end{cases}$$

ce schéma est explicite à un pas avec

$$\Phi(t, y, \Delta t) = f(t + \frac{\Delta t}{2}, y + \frac{\Delta t}{2} f(t, y))$$

Comme

- $\Phi(t, y, 0) = f(t, y)$ ce schéma est consistant d'ordre au moins 1.
 - $\frac{\partial \Phi}{\partial \Delta t}(t, y, \Delta t) = \frac{1}{2} \frac{\partial f}{\partial t}(t + \frac{\Delta t}{2}, y + \frac{\Delta t}{2} f(t, y)) + \frac{1}{2} f(t, y) \frac{\partial f}{\partial y}(y + \frac{\Delta t}{2} f(t, y))$
- $$\frac{\partial \Phi}{\partial \Delta t}(t, y, 0) = \frac{1}{2} \left[\frac{\partial f}{\partial t}(t, y) + f(t, y) \frac{\partial f}{\partial y}(t, y) \right] = \frac{1}{2} f^{[1]}(t, y)$$

D'où le schéma des (PM) est consistant d'ordre au moins 2.

Exercise. Mq qu'il ne peut pas être consistant d'ordre. Choisissez $f(t, y) = g(t)$ et revenez à la définition de $\xi(t, \Delta t)$

4.4.3 Stabilité des schémas à un pas

Definition 4.24. (S) est di stable pour une classe de fonction f , s'il existe une constante $S \geq 0$ indépendante de Δt tq:

Pour toute suite $(x_n)_{0 \leq n \leq N}$, $(\tilde{x}_n)_{0 \leq n \leq N}$, $(r_n)_{0 \leq n \leq N-1}$:

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

$$\tilde{x}_{n+1} = \tilde{x}_n + \Delta t \Phi(t_n, \tilde{x}_n, \Delta t) + r_n$$

on a la majoration

$$\underbrace{\max_{0 \leq n \leq N} \{ \| \tilde{x}_n - x_n \| \}}_{\substack{\text{erreur maximale} \\ \text{pour toutes les itérations}}} \leq S \{ \underbrace{\| \tilde{x}_0 - x_0 \|}_{\substack{\text{erreur initiale}}} + \sum_{j=0}^{N-1} \underbrace{\| \eta_j \|}_{\substack{\text{erreur ajoutée} \\ \text{à chaque itération}}} \}$$

Proposition 4.25. (Stabilité du schéma d'Euler explicite)

Si f est lipschitzienne en espace, alors le schema d'Euler explicite est stable.

Proof.

$$\| x_{n+1} - \tilde{x}_{n+1} \| \leq (1 + L\Delta t) \| x_n - \tilde{x}_n \| + \| \eta_n \|$$

Et on conclut pas le lemme de Gramwall discret. \square

Lemma 4.26. de Graomwall discret:

$(a_n)_{0 \leq n \leq N}$, $(b_n)_{0 \leq n \leq N-1}$ termes positifs et $h > 0$, $L > 0$ tq

$$a_{n+1} \leq (1 + Lh)a_n + b_n \quad n = 0, \dots, N-1$$

alors

$$a_n \leq e^{Lnh} a_0 + \sum_{j=0}^{n-1} e^{L(n-j-1)h} b_j$$

D'où

$$a_n \leq e^{nhL} (a_0 + n \max_{0 \leq j \leq n-1} |b_j|)$$

Proposition 4.27. Si la fonction Φ est telle que: il existe $\Lambda \geq 0$ tq:

$$\forall \Delta t \in [0, \Delta t_0], \forall t \in [t_0, t_0 + T - \Delta t] \forall y_1, y_2 \in \mathbb{R}^d$$

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

alors le schéma (S) est stable de constante de stabilité $S = e^{\Lambda T}$

4.4.4 Convergence des schémas à un pas explicite

Theorem 4.28. 1. Si la méthode (S) est stable et consistant, alors elle est convergente

2. Si la méthode (S) est stable et consistante d'ordre q , alors elle est convergente d'ordre q et pour toute suite $(x_n)_{0 \leq n \leq N}$ donnée par S , initialisée par $x_0 \in \mathbb{R}^d$, on a

$$\max_{0 \leq n \leq N} \|x(t_n) - x_n\| \leq (\|x(t_n) - x_0\| + CT\Delta t^q)$$

Où S est une constante de stabilité et C - constante de consistance.

CHAPTER 5

RÉSOLUTION APPROCHÉE D'ÉQUATIONS ORDINAIRES (EO) : $f(x) = 0$

5.1 Généralités et exemples

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, $I \subset \mathbb{R}^n$

$$(EO) \begin{cases} \text{chercher } x^* \in I \\ f(x^*) = 0 \end{cases}$$

5.1.1 Définitions

- (EO) est appelé *équation non linéaire* si f est affine, elle est dite *linéaire*.

$$f(x) = Ax + b \quad \text{où} \quad \begin{cases} A \in \mathcal{M}_m(\mathbb{R}) \\ b \in \mathbb{R}^n \end{cases}$$

- Tout $x^* \in I$ solution de (EO) est dite racine au zéro de f dans I .
- Si f est de classe \mathcal{C}^r avec $r \geq 1$ et x^* est racine de f , alors
 - x^* est dite racine simple $f'(x^*) \neq 0$
 - x^* est dite racine de multiplicité $p < r$ si $f^{(k)}(x^*) = 0$, $k = 0, \dots, p-1$ et $f^{(p)}(x^*) \neq 0$
 - lorsque $m = 1$, c'est cad $f : I \subset \mathbb{R} \rightarrow \mathbb{R}$, l'équation est dite scalaire.

5.1.2 Exemples

Schéma d'Euler implicite par EDO

$$(EI) \begin{cases} x' = f(t, x) \\ x(t_0) = x^0 \\ x_{n+1} = x_n + \Delta t f(t_{n+1}, x_{n+1}), n = 0, \dots, N-1 \\ x_0 = x^0 \end{cases}$$

À l'itération n , pour déterminer x_{n+1} , il faut résoudre l'équation $g_n(z) = 0$ où

$$g_n(z) = z - x_n + \Delta t f(t_{n+1}, z)$$

- x^0 donné
- Pour $n = 0, \dots, N-1$,
 - résoudre $\begin{cases} \text{chercher } z \in \mathbb{R} \\ z = x_n - \Delta t f(t_n + \Delta t, z) = 0 \end{cases}$
 - pour $x_{n+1} = z$

$$(CN) \begin{cases} x_{n+1} = x_n + \frac{\Delta t}{2} [f(t_n, x_n) + f(t_{n+1}, x_{n+1})], n = 0, \dots, N-1 \\ x_0 = x^0 \end{cases}$$

$$(HEUN) \begin{cases} \bar{x}_{n+1} = x_n + \Delta t f(t_n, x_n) \\ x_{n+1} = x_n + \frac{\Delta t}{2} [f(t_n, x_n) + f(t_{n+1}, \bar{x}_{n+1})] \\ x_0 = x^0 \end{cases}$$

5.2 Position correcte du problème (EO)

On se propose de vérifier si

- (EO) admet une solution
- Si oui, cette solution est-elle unique?
- La solution dépend continûment des données du problèmes (stabilité)
- La solution est suffisamment régulièrre.

Pour répondre à cela, on va se placer dans le cadre scalaire: $f : \mathbb{R} \rightarrow \mathbb{R}$

Proposition 5.1. Cas où $f(x) = 0$

- Si $I = [a, b]$ et f est telle que
 - $f(a)f(b) < 0$
 - f continue sur Ialors il existe $x^* \in I$ tq: $f(x^*) = 0$
- Si de plus f est strictement monotone, alors x^* est unique

Proof. • Théorème des valeurs intermédiaires

- unicité f est injective (où f bijective de $I \rightarrow f(I)$, $x^* = f^{-1}(0)$ est unique)

□

Proposition 5.2. Cas $x = g(x)$

Si g est telle que

- $g(I) \subset I$
- g continue

Alors , il existe $x^* \in I$ tq $g(x^*) = x^*$

- Si g contractante ($|g(x) - g(y)| < k|x - y|$ avec $0 \leq k < 1$) alors x^* est unique.

Remark 5.3. Si g est dérivable avec $|g'(x)| < j \ll 1 \forall x \in I$, alors g est contractante.

Remark 5.4. Lorsque la racine cherchée sera de multiplicité ≥ 1 . Il faudra faire attention!.

- Autrement dit, le problème sera difficile à résoudre numériquement si le zéro est une racine multiple.

5.3 Construction de schéma pour (EO)

5.3.1 Méthode de dichotomie

S'applique au cas $f(x) = 0$

Principe

$I = [a, b]$ tq $f(a) \cdot f(b) < 0$

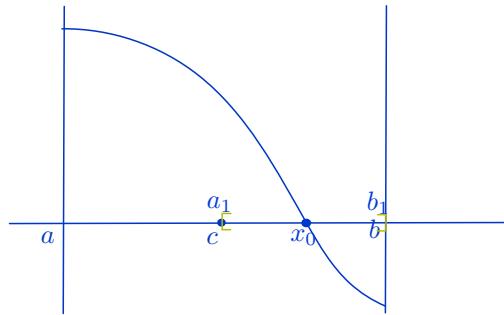


Figure 5.1: principe-de-methode-de-dichotomie

$$[a_0, b_0] = [a, b], c = \frac{a_0 + b_0}{2}$$

$$[a_1, b_1] = \begin{cases} [a_0, c] & \text{si } f(a_0)f(c) < 0 \\ [c, b_0] & \text{si } f(c)f(b_0) < 0 \end{cases}$$

Si $[a_n, b_n]$ est construit, on construit $[a_{n+1}, b_{n+1}]$, en prenant: $c = \frac{a_n + b_n}{2}$

$$[a_{n+1}, b_{n+1}] = \begin{cases} [a_n, c] & \text{si } f(a_n)f(c) < 0 \\ [c, b_n] & \text{sinon} \end{cases}$$

Algorithme

- Initialisation

- Soit a, b tq $f(a)f(b) < 0$ et ε tolérance donné
- Calculer $c = \frac{a+b}{2}$
- $k = 0$

- Itération: tant que $|f(c)| > \varepsilon$ et k pas trop grand, Faire

- Si $f(a)f(b) < 0$, poser $b = c$
- Sinon, poser $a = c$

Calculer $c = \frac{a+b}{2}$, $k = k + 1$. Fin Faire

Coût

:

- 1 dimension
- 1 évaluation de la fonction

Convergence

f est continue. Soit $(a_n), (b_n)$ générées

- $x^* \in [a_n, b_n]$
- $[a_{n+1}, b_{n+1}] \subset [a_n, b_n], \forall n$

- $b_n - a_n = \frac{b-a}{2^n}$
- $|a_n - x^*| \leq \frac{b-a}{2^n}, |b_n - x^*| \leq \frac{b-a}{2^n}$
- $\lim_{n \rightarrow \infty} a_n = x^*, \lim_{n \rightarrow \infty} b_n = x^*$

5.3.2 Méthode de fausse position

Principe

- remplacer dans dichotomie $c = \frac{a+b}{2}$ par c l'abscisse du point d'intersection avec l'axe des abscisses de la droite passante par $(a, f(a)), (b, f(b))$

$$c = b - \frac{b-a}{f(b)-f(a)} f(b)$$

Algorithme

Remplacer $c = \frac{a+b}{2}$ par $c = b - \frac{b-a}{f(b)-f(a)} f(b)$ dans dichotomie

- Initialisation
 - Soit a, b tq $f(a)f(b) < 0$ et ε tolérance donné
 - Calculer $c = b - \frac{b-a}{f(b)-f(a)} f(b)$
 - $k = 0$
 - Itération: tant que $|f(c)| > \varepsilon$ et k pas trop grand, Faire
 - Si $f(a)f(b) < 0$, poser $b = c$
 - Sinon, poser $a = c$
- Calculer $c = b - \frac{b-a}{f(b)-f(a)} f(b)$, $k = k + 1$. Fin Faire

Coût

:

- 1 dimension, 1 produit
- 1 évaluation de la fonction

Convergence

(Fausse position) Soit $f \in \mathcal{C}^2([a, b])$, $f(a)f(b) < 0$. Si f'' n'a aucune racine dans $I = [a, b]$, alors une des suites (a_n) ou (b_n) demeure constante.

Proposition 5.5. Si f est $\mathcal{C}^2([a, b])$, $f(a)f(b) < 0$ et f'' n'a aucune racine sur $[a, b]$. Soient $(a_n), (b_n)$ générées.

- Si (b_n) est constante, alors (a_n) converge linéairement vers x^* racine de f et on a

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

- Si (a_n) est constante (b_n) converge linéairement vers x^* et

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

5.4 Méthode d'itération (ou du point fixe)

5.4.1 Principe

Elle est adaptée aux problèmes

$$(EO) \begin{cases} \text{Chercher } x^* \in \mathbb{R} \\ f(x^*) = x^* \end{cases}$$

Remark 5.6. Une équation $f(x^*) = 0$ devra être transformée pour rentrer dans ce cadre ($g = ?$)
On cherche x^* comme $\lim_{n \rightarrow \infty} x_n$ où

$$(S) \begin{cases} x_{n+1} = y(x_n), n = 0, \dots \\ x_0 \text{ donné} \end{cases}$$

5.4.2 Interpolation géométrique

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

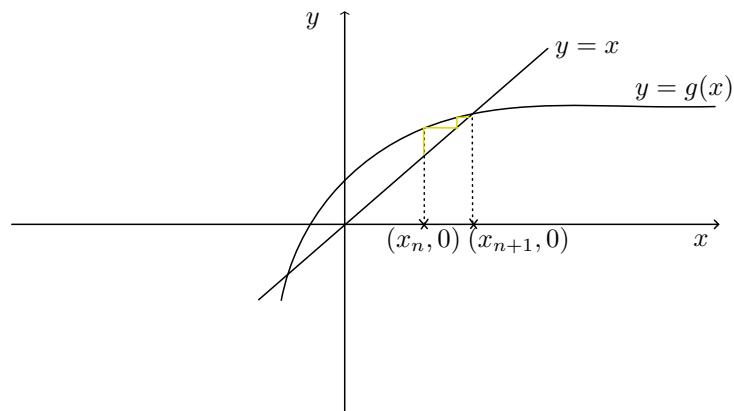


Figure 5.2: Interpolation géométrique

5.4.3 Algorithme

Initialisation

- Choisir x_0 et ε donné $\varepsilon > 0$
- Pour
 - $k = 0$
 - $r_k = \varepsilon + 1$

Itérations

- Tant que $r_k > \varepsilon$ et k pas trop grand faire
 - $x_{k+1} = g(x_k)$
 - $r_{k+1} = |x_{k+1} - x_k|$

– $k = k + 1$

- Fin faire

Coût:

1 évaluation de la fonction

Inconvenient:

Choix de x^*

5.4.4 Convergence

Proposition 5.7. Soit $I \subset \mathbb{R}$ tq $g(I) \subset I$, alors g admet un point fixe x^* dans \mathbb{R}

- Si $\max_{x \in I} |g'(x)| = k < 1$, alors le point fixe est unique
- Si $\max_{x \in I} |g'(x)| = k < 1$, et de plus, $x_0 \in I$ et $(x_n)_n$ générée par (S) , alors
 - $|x^* - x_{n+1}| < k|x^* - x_n|$
 - $(x_n)_n$ converge vers x^*
 - $k_1 = \lim_{n \rightarrow \infty} \frac{x^* - x_{n+1}}{x^* - x_n} = g'(x^*)$ et $k_1 \leq k$

Proof. • **Existence:** $I = [a, b]$. Posons $f(x) = f(x) - x$, on a: Comme $g(I) \subset I$, comme $g(a) \in [a, b]$ et $g(b) \in [a, b]$ on a: $g(a) \geq a$, $g(b) \leq b$, d'où $f(a) = g(a) - a \geq 0$, $f(b) = g(b) - b \leq 0$. D'après le théorème de VF, $\exists x^* \in [a, b]$ tq $f(x^*) = 0$ i.e $g(x^*) = x^*$

- **Unicité:** Supposons $\exists x^*, x^{**}$ tq $g(x^*) = x^*$ et $g(x^{**}) = x^{**}$ et $x^{**} \neq x^*$, alors

$$\begin{aligned} |x^{**} - x^*| &= |g(x^{**}) - g(x^*)| \\ &\leq |g'(x^* + o(x^{**} - x^*))||x^{**} - x^*| \\ &\leq \max_{n \in I} |g'(n)||x^{**} - x^*| \end{aligned}$$

$$|x^{**} - x^*| < K|x^{**} - x^*| \text{ avec } 0 < K < 1$$

i.e $1 < K < 1$ - absurde!

- **Convergence:**

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

D'où $|x_{n+1} - x^*| \leq |g(x_n) - g(x^*)| \leq K|x_n - x^*|$. D'où $|x_{n+1} - x^*| \leq K|x_n - x^*|$ donc

$$|x_n - x^*| \leq K^n|x_0 - x^*|$$

Comme $0 < K < 1$ on a $\lim_{n \rightarrow \infty} x_n = x^*$

- **Ordre de convergence:**

$$\begin{aligned} x_{n+1} - x^* &= g(x_n) - g(x^*) \\ &= g'(x^* + \theta(x_n - x^*))(x_n - x^*) \quad 0 < \theta < 1 \end{aligned}$$

D'où

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

Comme g est de classe C^1 , on a $\lim_{n \rightarrow \infty} g'(x^* + \theta(x_n - x^*)) = g'(x^*)$ Donc

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

□

Remark 5.8. Si $g'(x^*) \neq 0$, on a une convergence linéaire (d'ordre 1) avec pour constante asymptotique $K_1 = |g'(x^*)|$

Que se passe-t-il si $g'(x^*) = 0$?

Soit $p \geq 1$, le plus petit entier tel que $g^{(p)}(x^*) \neq 0$

Example 5.9.

$$\left\{ \begin{array}{l} x^{n+1} = g(x^n) \\ \end{array} \right.$$

avec

$$g(x) = e^x - 1 - x$$

$$\text{I. } g(x) = e^x - 1 - x - \frac{x^2}{2}$$

Rq: $x^* = 0$

$$g'(0) = 0, g''(0) = 1, \text{ alors } p = 2$$

$$\text{II. } g'(0) = 0, g''(0) = 0, g'''(0) = 1 \neq 0, \text{ alors } p = 3$$

Si $g \in C^p$, alors:

$$g(x_n) = g(x^*) + (x_n - x^*)g'(x^*) + \sum_{k=2}^{p-1} \frac{g^{(k)}(x^*)}{k!} (x_n - x^*)^k + \frac{(x_n - x^*)^p}{p!} g^{(p)}(x^* + \theta(x_n - x^*))$$

Comme $g'(x^*) = \dots = g^{(p-1)}(x^*) = 0$, on a: $g(x^*) = x^*, g(x_n) = x_{n+1}$

$$x_{n+1} - x^* = (x_n - x^*)^p \frac{g^{(p)}(x^* + \theta(x_n - x^*))}{p!}$$

D'où

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

Donc $x_n \rightarrow x^*$ à l'ordre p .

5.5 Méthode de Newton

Principe

Adaptée pour le problème

$$(EO) \begin{cases} \text{Cherche } x^* \in I \\ f(x^*) = 0 \end{cases}$$

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

On remplace f par son polynôme de Taylor d'ordre 1 au voisinage de x_n

$$f(x) \approx f(x_n) + f'(x_n)(x - x_n) \equiv P_f(x)$$

On trouve $f(x_n) + f'(x_n)(x_{n+1} - x_n) = 0$. D'où $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$, c'est à dire $x_{n+1} = g(x_n)$ avec $g(x) = x - \frac{f(x)}{f'(x)}$

Comment calculer un point fixe donné

```
1 def PointFixe(g, x_0, eps, IterMax):
2     ...
```

```

1 def Newton(f, f_prime, x0, eps, IterMax):
2     g = lambda x: x - f(x)/f_prime(x)
3     return PointFixe(g, x0, eps, IterMax)

```

Convergence

On adapte les résultats de la méthode du point fixe, ici:

Comme $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$, n'est rien d'autre que

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

Remark 5.10.

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

On a:

$$g'(x) = 1 - \frac{f'(x)^2 - f(x)f''(x)}{(f'(x))^2}$$

D'où

$$g'(x^*) = 1 - \frac{f'(x^*)^2 - 0}{f'(x^*)^2} = 0$$

D'où la méthode de Newton. Pour un bon choix de x_0 converge. Et est au moins d'ordre 2 pour une racine simple.

Remark 5.11. Si $f(x^*) = 0, f'(x^*) = 0, \dots, f^{(m)}(x^*) \neq 0$

Example 5.12. $f(x) = (x - 1)^3$, $x^* = 1$, on a: $f(1) = 0, f'(1) = 0, f''(1) = 0, f^{(3)}(1) \neq 0$, i.e $m = 3$

$$\begin{cases} x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \\ x_0 \text{ donnée} \end{cases}$$

On aura convergence d'ordre 1 i.e $\lim_{n \rightarrow \infty} \frac{x_{n+1} - x^*}{x_n - x^*} = K_1 \neq 0$ avec $K_1 = 1 - \frac{1}{m} = \frac{2}{3}$
I.e x^* est racine d'ordre m de f d'où $f(x) = (x - x^*)^m h(x)$ avec $h(x^*) \neq 0$