MACS 205: Analyse numérique
notes de cours (succintes)

A. Sabourin, Olivier Fercoq, François Portier

11 février 2021

Table des matières

Ce cours d'analyse numérique (calcul scientifique) s'appuie principalement sur les ouvrages de Quarteroni et al. [2008] et Demailly [2012]. Les deux premiers chapitres de ce document sont issus des notes de cours rédigées par Baptiste Portenard, que nous remercions chaleureusement.

Fonctionnement du cours

Les cours de Macs 205 (a) et (b), sont en pratique regroupés en un seul bloc Macs 205. Celui-ci est divisé en trois partie, correspondant chacune à un intervenant :

(1) Fondements du calcul scientifique, interpolation polynomial et intégration numérique → Anne Sabourin

(2) Méthodes numériques pour équations différentielles → Olivier Fercoq

(3) Méthodes de Monte-Carlo → François Portier.

Déroulé du cours : alternance de cours, TD et TP. La partie 'implémentation' des méthodes prend un place relativement importante (≈ la moitié du temps). Le langage utilisé est R.

Evaluation Elle comprend une partie théorique (examens écrits) et une partie pratique (mini-projets) Pour des raisons administratives, deux notes indépendantes doivent être attribuées. Chaque note est calculée sur le barême suivant, à partir de 2 examens écrits et 2 mini-projets :

- MACS 205-a :
  - mini-projet sur la partie (1) : coeff 1.
  - mini-projet sur la partie (2) : coeff (1/2)
  - examen écrit sur les parties (1) et (2) : coeff (1)
- MACS 205-b
  - mini-projet sur la partie (2) : coeff 1/2.
  - examen écrit sur la partie (3) : coeff 2

Chapitre 1

Fondements du calcul scientifique

1.1 Exemples, sources d'erreur, problèmes bien posés, conditionnement

But de l'analyse :
Résoudre un problème mathématique de type

$$F(x, d) = 0,$$

où F est une fonction à valeurs réelles qui représente le « modèle mathématique », x est l'inconnue recherchée et d représente l'ensemble des données du problèmes.
On note x_0 une solution en x de ce problème. On a recourt à des méthodes numériques lorsque x_0 n'a pas d'expression explicite.

Méthode générale :
On construit un algorithme produisant une suite de solutions approchées $(x_n)_{n\geqslant 1}$ du problème approché

$$F_n(x, d_n) = 0$$

dans l'espoir que la suite (x_n) tende vers une solution x_0 du problème initial,

$$x_n \to x_0.$$

Pour l'instant on n'a pas précisé dans quel sens cette convergence doit avoir lieu. La notion de convergence dépendra généralement du choix d'une norme dans l'espace des solutions.

1.1.1 Sources d'erreur

La résolution du problème initial concret (que l'on appellera *problème physique* passe par la formulation successive d'un *problème mathématique* (censé approcher le problème physique), puis d'un *problème numérique* approché, dont la résolution entraîne à son tour des *erreurs de mesure et d'arrondi*. Ces différentes étapes sont illustrées dans l'exemple suivant

Exemple 1.1 (Problème du robinet)**:**
On considère une baignoire de volume V et un robinet de débit $q(t), t > 0$. On désire connaitre x le temps nécessaire pour remplir une baignoire.

Problème physique On peut représenter mathématiquement le problème physique sous la forme

$$F_{phys}(x, d_{phys}) = 0$$

avec les paramètres suivants

$$d_{phys} = (q : \mathbb{R}_+ \to \mathbb{R}_+,\ V > 0)$$

$$F_{phys}(x, V, q) = V - \int_0^x q(t)dt$$

Etape de modélisation : Modification des données du problèmes. On suppose la baignoire est un parallélépipède rectangle (*i.e.,* un pavé), de sorte que $V \simeq l.L.h$.

Problème mathématique Le problème devient :

$$F(x, d) = 0, \quad d = \{l, L, h, q\}$$

où F s'écrit

$$F(x, l, L, h, q) = l.L.h - \int_0^x q(t)dt.$$

Du fait de l'étape de modélisation, on a $x_0 \neq x_{phys}$.

Etape de discrétisation : Modification des données du problèmes

$$\int_0^x q(t)dt \simeq \sum_{k=0}^{\lfloor x.n \rfloor} \frac{1}{n} q(\frac{k}{n})$$

Problème numérique l'équation définissant le problème numérique est à présent

$$F_n(x, d_n) = 0, \quad d_n = \{q(\frac{k}{n}), L, h, q\},$$

où :

$$F_n(x, d_n) = l.L.h - \sum_{k=0}^{\lfloor x.n \rfloor} \frac{1}{n} q(\frac{k}{n}).$$

Erreur de mesure, arrondi La mesure des données est soumise a des approximations, par exemple pour calculer un débit on moyenne sur une petite durée d'écoulement. On note avec un chapeau les variables mesurées et on obtient la solution $\widehat{x_n}$ du problème approché

$$\widehat{F_n}(x, \widehat{d_n}) = \widehat{l}.\widehat{L}.\widehat{h} - \sum_{k=0}^{\lfloor x.n \rfloor} \frac{1}{n} \widehat{q}(\frac{k}{n})$$

Dans ce cours, on ne s'intéressera pas à l'erreur venant de l'étape de modélisation. Autrement dit, on considérerar que x_0 est la « vraie » solution et on analysera les erreurs venant des étapes suivantes de l'exemple ci-dessus.

Définition 1.1.1 (Erreurs)**.** *On définit les différentes erreurs suivantes*

- *Erreur totale* $= \widehat{x_n} - x_{phys}$
- *Erreur d'arrondi* $= \widehat{x_n} - x_n$

- *Erreur de troncature* $= x_n - x_0$
- *Erreur de modèle* $= x_0 - x_{phys}$.

On appelle erreur de calcul *la somme des erreurs d'arrondi et de troncature (i.e.* $\widehat{x_n} - x_0$*)*

Comme $\widehat{x_n} - x_{phys} = (\widehat{x_n} - x_n) + (x_n - x_0) + (x_0 - x_{phys})$, on en déduit que l'erreur totale est la somme des différentes autres erreurs de la définition précédente. En particulier,

$$\widehat{x_n} - x_{phys} = \underbrace{\widehat{x_n} - x_0}_{\text{erreur de calcul}} + \underbrace{x_0 - x_{phys}}_{\text{erreur de modèle}}$$

Et on cherchera à contrôler le terme d'erreur de calcul dans la décomposition ci-dessus, en fonction de

- $\delta d = \widehat{d_n} - d_n$: erreur d'arrondi.
- l'erreur de troncature des données $d_n - d$ (dans l'exemple, $d_n - d = \sum_{k=0}^{\lfloor x.n \rfloor} \frac{1}{n} q(\frac{k}{n}) - \int_0^x q$).

La sensibilité d'une méthode numérique par rapport à δd est résumée par la notion de *stabilité* et de *conditionnement* (voir le paragraphe 1.1.3). La sensibilité par rapport à $d_n - d$ est résumée par la notion de *consistance* (voir le paragraphe 1.2).

1.1.2 Problème bien posé

Définition 1.1.2 (Problème bien posé)**.** *Un problème mathématique*

$$F(x, d) = 0$$

est bien posé si

- *Pour toutes données d il existe une unique solution* $x_0 = G(d)$*. On appelle G la fonction résolvante.*
- *G est continue*

Exemple 1.2:
On considère le problème suivant

$$F(x, d) = x^2 - 2xd + 1$$

dont on cherche les solutions (x_1, x_2) dans $\mathbb{R}$.
Le discriminant réduit est $\Delta = d^2 - 1$
Si $|d| < 1$ alors le problème est mal posé.
sinon on a

$$G(d) = (d - \sqrt{\Delta}, d + \sqrt{\Delta})$$
$$G(d) = (d - \sqrt{d^2 - 1}, d + \sqrt{d^2 - 1})$$

et G est continue sur $\mathbb{R} \setminus]-1, 1[$

1.1.3 Stabilité

Définition 1.1.3 (Stabilité)**.** *Un problème* $F(x, d) = 0$ *est stable si la résolvante* $G(d)$ *est localement lipschitzienne.*

$$\forall d, \exists \eta > 0, \exists K_0(d) \text{ tel que } \|h\| < \eta \Rightarrow \|x(d + h) - x(d)\| \leqslant K_0(d)\|h\| \tag{1.1}$$

où $x(d) = G(d)$

Proposition 1.1.4
La condition (1) de stabilité est équivalente à

$$\lim_{\epsilon \to 0} sup_{\|h\|<\epsilon} \left\| \frac{x(d+h)-x(d)}{h} \right\| < +\infty \tag{1.2}$$

Définition 1.1.5 (Conditionnement)**.** *On définit le conditionnement d'un problème stable comme la meilleure constante de Lipschitz locale.*

$$K_{abs} = \lim_{\epsilon \to 0} \sup_{\|h\|<\epsilon} \left\| \frac{x(d+h)-x(d)}{h} \right\| \tag{1.3}$$

On définit également le conditionnement relatif

$$K_{rel} = K_{abs} \frac{||d||}{||x(d)||} \tag{1.4}$$

Exemple 1.3:
Soit $A \in \mathcal{M}_{n,m}(\mathbb{R})$, $y \in \mathbb{R}^n$ et $y \in \mathbb{R}^m$. On considère le problème

$$F(x,d) = Ax - y \text{ avec } d = (A,y)$$

Le problème mathématique (PM) a une unique solution si A est inversible, alors $x_0 = A^{-1}(y) = x(A,y)$. Avec la norme $\| \ \|_2$

$$x(d) = A^{-1} = \frac{1}{\det A}(\mathrm{Com}A)^T y$$

$$\|A\|_2 = \sqrt{\mathrm{Tr}(AA^T)}$$

et donc G est continue en (A,y). On en déduit que le problème est bien posé.
Maintenant on suppose que A est fixé. La donnée est donc $d = y$. On a

$$\begin{aligned} \frac{||x(d+h)-x(d)||}{\|h\|} &= \frac{||A^{-1}(y+h)-A^{-1}(y)||}{\|h\|} \\ &= \frac{||A^{-1}(h)||}{\|h\|} \\ &= \left\| A^{-1}\left(\frac{h}{\|h\|}\right) \right\| \end{aligned}$$

d'où

$$K_{abs} = \sup_{||u||=1} \left\| A^{-1}(u) \right\| = |||A^{-1}|||$$

Si A est symétrique, on note ses valeurs propres $(\lambda_1, ..., \lambda_n)$. Sans perte de généralité on peut les ordonner en valeur absolue $|\lambda_1| \geqslant ... \geqslant |\lambda_n|$. On obtient alors

$$|||A^{-1}||| = \frac{1}{|\lambda_n|}$$

Mais également

$$K_{rel} = K_{abs} \frac{||d||}{||x||} = K_{abs} \frac{||y||}{||A^{-1}(y)||} = \frac{1}{|\lambda_n|} \frac{||Ax||}{||x||}$$

$$K_{rel} \leqslant \frac{1}{|\lambda_n|} \sup_{||x||=1} \|Ax\| = \frac{|\lambda_1|}{|\lambda_n|}$$

1.1.4 Calcul de conditionnement lorsque G est différentiable

Définition 1.1.6 (Différentielle)**.** *G est différentiable en y si il existe A linéaire tel que*

$$\forall d \in \mathbb{R}^n, G(y+h) = G(y) + A(h) + o(\|h\|)$$

On note $A = \mathrm{d}G(y)$ la différentielle de G en y.

Proposition 1.1.7
On a le conditionnement

$$K_{abs} = |||\mathrm{d}G||| \tag{1.5}$$

et

$$K_{rel} = |||\mathrm{d}G|||\frac{||d||}{||G(d)||}$$

Démonstration. On calcule le conditionnement

$$\begin{aligned}\frac{||x(d+h) - x(d)||}{\|h\|} &= \frac{||x(d) + \mathrm{d}G(h) + o(\|h\|) - x(d)||}{\|h\|} \\ &= \frac{||\mathrm{d}G(h)||}{\|h\|} + o(1)\end{aligned}$$

d'où

$$\limsup_{\|h\|\to 0}\left(\frac{||x(d+h) - x(d)||}{\|h\|}\right) = |||\mathrm{d}G|||$$

et donc également

$$K_{rel} = K_{abs}\frac{||d||}{||x||} = |||\mathrm{d}G|||\frac{||d||}{||G(d)||}$$

■

1.2 Convergence des méthodes numériques

1.2.1 Définitions

On suppose que le (PM) : $F(x,d) = 0$ est stable et bien posé.
Une méthode numérique est une suite de problème approchés, i.e. $F_n(x,d_n) = 0$ de solution x_n.

Définition 1.2.1. *Une méthode numérique est convergente si*

$$\forall d \in \mathbb{N}, \lim_{\substack{n\to\infty \\ d_n \to d}} x_n(d_n) = x(d) \tag{1.6}$$

En pratique la convergence est difficile à vérifier directement ; Il est parfois plus facile de vérifier les propriétés de consistance et de stabilités définie ci-dessous.

Définition 1.2.2 (Consistance)**.** *Une méthode numérique est consistante si*

$$\forall d \in \mathbb{N}, F_n(x(d), d) \to 0 \text{ lorsque } n \to \infty. \tag{1.7}$$

où $x(d)$ est solution du problème $F(x,d) = 0$.

La stabilité d'une méthode numérique est définie comme celle d'un problème mathématique à la section 1.1.3. Attention cependant au fait que la condition suivante est uniforme en n

Définition 1.2.3 (stabilité d'une méthode numérique)**.** *Une méthode numérique $F_n(x, d_n)$ est stable si pour tout d il existe $K_0(d)$ et δ tels que :*

$$\forall n,\ \sup_{\|h\|<\delta} \|x_n(d+h) - x_n(d)\| \leq K_0(d)\|h\|.$$

L'intérêt de telles définitions est le suivant :

Sous des hypothèses raisonnables, si une méthode est consistante et stable, alors elle est également convergente.

Exemple 1.4 (Convergence dans le cas où F_n est « suffisamment régulière »)**:**
On suppose que pour tout x, n, d,

1. $\frac{\partial F_n}{\partial x}(x, d)$ existe et est inversible.
2. absence de 'plateau' :
$$M = \sup_{x,d,n} \left\| \left[\frac{\partial F_n}{\partial x}(x, d) \right]^{-1} \right\| < +\infty$$
(intuitivement : F_n n'est pas trop 'plate').
3. La méthode numérique est stable et consistante.

On montre que sous ces hypothèses, la méthode est convergente.

Preuve : exercice.

1.3 Erreurs d'arrondis

(N.B. : arrondis = termes « δd » dans l'analyse de la stabilité).

1.3.1 Représentation en base β

Soit $\beta \geq 2 \in \mathbb{N}$. Tout nombre réel x s'écrit

$$x = (-1)^s \sum_{k=-\infty}^{n} x_k \beta^k,$$

pour un certain $n \in \mathbb{N}$, avec $s \in \{-1, 1\}$ (le signe) et $x_k \in [0, \ldots, \beta - 1]$. On note $[x]_\beta = x_n x_{n-1} \ldots x_1 x_0 \cdot x_{-1} x_{-2} \cdots$ la représentation en base β.

1.3.2 Nombres à virgule flottante ("float")

Souvent $\beta = 2$ ou $\beta = 16$. On veut pouvoir représenter un panel le plus grand possible de nombres en base β, avec N cases mémoires seulement.

Méthode naïve : utiliser directement la représentation en base β ci-dessus, tronquée à k chigffres après la virgules : $\to$ réel max $= \beta^{N-k} - 1$: pas grand !

Principe des flottants : faire 'flotter' la virgule et réserver une case pour l'exposant e.

$$x = (-1)^s \times 0.\underbrace{a_1 a_2 \dots a_t}_{\text{taille } t} \times \beta^e$$

- t : nombre de chiffres significatifs
- $a_i \in \{0, \dots, \beta - 1\}$
- '$a_1 a_2 \dots a_t'$: mantisse.
- e : un entier : l'exposant. Contrainte sur e : $L \leq e \leq U$. (L et U sont des entiers dépendent de la taille réservée pour e.
- On impose : $a_1 \neq 0 \rightarrow$ unicité de la représentation.

exemples :

- systeme 32 bits ('float') : s$\mapsto$ 1 bit ; $e \mapsto$ 8 bits, $m \mapsto$ 23 bits.
- systeme 64 bits ('double') : s$\mapsto$ 1 bit ; $e \mapsto$ 11 bits, $m \mapsto$ 52 bits.

on appelle $\mathbb{F}(\beta, t, L, U)$ l'ensemble des nombres flottants représentables de cette manière.

!!! c'est un ensemble fini ($\neq \mathbb{R}$) !!!

réels min et max : tout flottant x vérifie

$$0.1\beta^L \leq |x| \leq 0.(\beta-1)(\beta-1)\dots(\beta-1) \times \beta^U = \beta^U(1 - \beta^{-t})$$

epsilon machine

$$\epsilon = \min\{x > 0 : 1 + x \in \mathbb{F}(\beta, t, L, U)\}.$$

Puisque $1 = 0.\underbrace{1\dots0}_{\text{taille } t} \times \beta$, le nombre suivant est $1 + \epsilon = 0.\underbrace{1\dots01}_{\text{taille } t} \times \beta$.
D'où $\epsilon = (1 + \epsilon) - 1 = 0.0\dots01 \times \beta = \beta^{1-t}$.

Dans `R` : Toutes ces constantes sont acessibles dans la liste `.Machine`

1.3.3 Opérations sur les flottants

On note $fl(x)$ la représentation machine d'un réel x. Pour toute opération arithmétique $\cdot \in \{+, -, \times, \div\}$, on note $\boxdot$ l'opération machine correspondante, *i.e.,* son analogue dans le systeme des nombres flottants. Plus précisément

$$x \boxdot y = fl[fl(x) \boxdot fl(y)].$$

Sur les machines suivant la dernière norme ISO en vigueur, IEE559 (*i.e.,* toutes les machines), on est assuré que

$$\left| \frac{x \boxdot y \ - \ x \cdot y}{x \cdot y} \right| \ \leq \ \frac{\epsilon}{2},$$

où ϵ = epsilon machine.

Chapitre 2

Interpolation Polynomiale

Enjeux de ce chapitre

- Représenter à faible coût des fonctions définies sur un intervalle borné.
- Avoir des garanties sur la qualité de la représentation en terme de norme de l'erreur
- mathématiquement : étudier la stabilité, la convergence des méthodes d'interpolation
- **Objectif pour les étudiants :** savoir implémenter une méthode d'interpolation et savoir évaluer l'erreur commise.

2.1 Interpolation de Lagrange

2.1.1 Existence, unicité

Soit $a < b$ deux réels. On considère f une fonction de $[a, b]$ dans $\mathbb{R}$.
On cherche a construire un polynôme p de degré inférieur à $n \in \mathbb{N}$ tel que pour $n+1$ points d'évaluations $a \leqslant x_0 < x_1 < ... < x_n \leqslant b$ on ait

$$\forall i \in [\![0; n]\!], p(x_i) = f(x_i)$$

Définition 2.1.1 (Base de Lagrange)**.** *Pour les points d'évaluations précédents, on définit les $n+1$ polynômes suivants*

$$\forall i \in [\![0; n]\!], l_i(x) = \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}$$

Soit $\mathbb{P}_n$ l'espace vectoriel (de dimension $n+1$) des polynômes réels de degré n.

Proposition 2.1.2
Les polynômes de la base de Lagrange vérifient les points suivants :

1. $\forall i \in [\![0; n]\!], l_i \in \mathbb{P}_n$
2. $\forall i, j \in [\![0; n]\!], l_i(x_j) = \delta_{ij}$
3. *Ils constituent une base de $\mathbb{P}_n$*

Définition 2.1.3 (Interpolée de Lagrange)**.** *On définit l'interpolation de f pour les $n+1$ points d'évaluations précédents comme le polynôme vérifiant*

$$L_n f = \sum_{i=0}^{n} f(x_i) l(x_i)$$

L_n est un opérateur linéaire, i.e. $L_n(f+g) = L_nf + L_ng$

Proposition 2.1.4
De la propriété précédente il vient que $L_nf \in \mathbb{P}_n$ et

$$\forall i \in [\![0; n]\!], f(x_j) = L_nf(x_i)$$

De plus, comme les polynômes (l_i) forment une base de $\mathbb{P}_n$, on en déduit que L_nf existe et est l'unique polynôme de degré au plus n a satisfaire cette propriété.

2.1.2 Erreur d'interpolation de Lagrange

Définition 2.1.5 (Polynôme nodal)**.** *Pour $n+1$ points d'évaluations $(x_0, ..., x_n)$, on définit le polynôme nodal de degré $n+1$ comme*

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

Lemme 2.1.6
Si $g : [a, b] \to \mathbb{R}$ est p fois dérivable sur $[a, b]$ et si on dispose de $x_0 < ... < x_n$ dans $[a, b]$ tels que $\forall i \in [\![0; n]\!] g(x_i) = 0$, alors

$$\exists y \in]a, b[\text{ tel que } g^{(p)}(y) = 0$$

Démonstration. Par récurrence sur n : Soit H_p la propriété de l'énoncé au rang p.
Pour H_1 est vraie d'après le théorème de Rolle.
Supposons H_{p-1} vraie. Soient $x_0, \ldots, x_p$ comme dans l'énoncé. D'après le théorème de Rolle appliqué à g entre x_i et x_{i+1} , $\exists (c_0, ..., c_{p-1})$ avec $c_i \in]x_i, x_{i+1}[$ tel que $g'(c_i) = 0$. Par hypothèse de récurrence appliquée à g', il existe $y \in]c_0, c_{p-1}[$ tel que $(g')^{(p-1)}(y) = 0$. Ainsi $g^{(p)}(y) = 0$ et $y \in]a, b[$, ce qui démontre H_p et achève la récurrence. ■

Théorème 2.1.7
Si f est $(n+1)$ fois dérivable sur $[a, b]$ alors

$$\forall x \in [a, b], \exists y_x \in I_x = [min(x, x_0), max(x, x_n)] \subset [a, b]$$

$$\text{tel que } E_n(x) = f(x) - L_nf(x) = \frac{1}{(n+1)!}\omega_{n+1}(x)f^{(n+1)}(y_x)$$

Démonstration. On pose $x_{n+1} = x$ et on note $q_{n+1} = L_{n+1}f - L_nf \in \mathbb{P}_{n+1}$. On remarque que $q_{n+1}(x_0) = ... = q_{n+1}(x_n) = 0$ et donc q_{n+1} possède $n+1$ racines. Comme q_{n+1} est dans $\mathbb{P}_{n+1}$ on en déduit que

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

Or par définition de $L_{n+1}f$, en $X = x_{n+1} = x$ on a $L_{n+1}f(x) = f(x)$ et donc il vient que

$$E_n(x) = f(x) - L_nf(x) = C\omega_{n+1}(x)$$

On pose désormais $g = f - L_{n+1}f$ qui s'annule en $x_0, ..., x_{n+1}$. Comme de plus g est $n+1$ fois dérivable on peut appliquer le lemme 2.1.6, d'où l'existence de $y_{x_{n+1}} = y_x$ tel que $g^{n+1}(y_x) = 0$. Or,

$$\begin{aligned} g^{(n+1)}(y) &= f^{(n+1)}(y) - (L_n f - C\omega_{n+1})^{(n+1)}(y) \\ &= f^{(n+1)}(y) - C(n+1)!. \end{aligned}$$

Pour conclure, remarquons que

$$g^{(n+1)}(y_x) = 0 \Rightarrow C = \frac{f^{(n+1)}(y_x)}{(n+1)!}.$$

■

Corollaire 2.1.8
On déduit du théorème précédent

$$\|f - L_n f\|_{[a,b],\infty} \leqslant \frac{1}{(n+1)!} \|\omega_{n+1}\|_{[a,b],\infty} \left\|f^{(n+1)}\right\|_{[a,b],\infty}$$

2.1.3 Contre exemple de Runge

Le membre de droite dans le corollaire 2.1.8 fait intervenir la norme du polynôme nodal ω_{n+1}. Malheureusement, cette dernière « explose » pour les grandes valeurs de n, contrairement à l'intuition qui suggère qu'une interpolation avec des polynômes de degré élevé fournirait une meilleure approximation.

Exemple 2.1:
On pose sur $[-1, 1]$

$$f(x) = \frac{1}{1 + 5x^2}$$

Et on considère les noeuds equi - répartis $x_i = a + i\frac{b-a}{n}$ La figure 1. représente f et son interpolée de Lagrange.

2.1.4 Stabilité

En pratique on évalue les $f(x_i)$ par des valeurs approchées $\widehat{f}(x_i)$. On cherche à contrôler $||L_n\widehat{f} - L_n f||$ par rapport à $||\delta f|| = ||f - \widehat{f}||$. Ici et dans la suite, on choisit comme norme sur les fonctions bornées $[a, b] \to \mathbb{R}$:

$$\|h\| = \|h\|_{\infty,[a,b]} = \sup_{x\in[a,b]} |h(x)|.$$

En reprenant la définition du conditionnement on obtient

$$\begin{aligned} K_{abs}(f) &= \sup_{||\delta f||\to 0} \left\| \frac{L_n(f + \delta f) - L_n(f)}{\delta f} \right\| \\ &= \sup_{||\delta f||\to 0} \left\| \frac{L_n(\delta f)}{\delta f} \right\| \\ &= \sup_{||\delta f||=1} \|L_n(\delta f)\| \end{aligned}$$

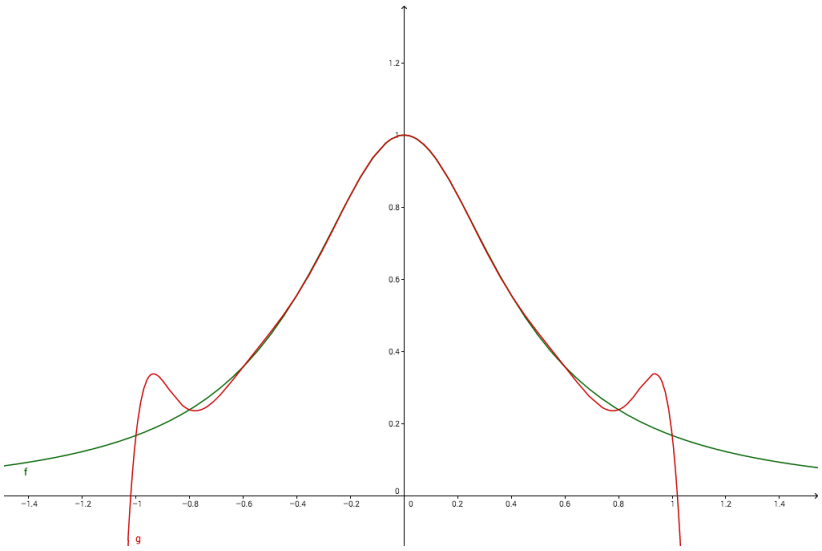


FIGURE 2.1 – f et son interpolée pour $n = 10$

Et on en déduit le résultat suivant

$$K_{abs}(f) = |||L_n|||.$$

(rappel : $|||L_n|||$ est la « norme de opérateur » de L_n, en prenant la norme $\|\cdot\|_{\infty,[a,b]}$ sur les fonctions continues sur $[a,b]$. Autrement dit $|||L_n||| = \sup_{\|f\|_{\infty,[a,b]}=1} \|L_n f\|$.

Calcul de $|||L_n|||$:
Dépend de $X^n = (x_0, ..., x_n)$
Soit g une fonction de $[a,b]$ dans $\mathbb{R}$

$$\begin{aligned}\|L_n g\| &= \sup_{x\in[a,b]} \left|\sum_{i=0}^{n} g(x_i) l_i(x)\right| \\ &\leqslant \sup_{x\in[a,b]} \sum_{i=0}^{n} |g(x_i)|\,|l_i(x)| \\ &\leqslant \|g\|_{[a,b]} \sup_{x\in[a,b]} \sum_{i=0}^{n} |l_i(x)|\end{aligned}$$

Définition 2.1.9 (Constante de Lebesgue)**.** *Pour $X^n = (x_0, ..., x_n)$ donné on définit la constante de Lebesgue*

$$\Lambda_n = \sup_{x\in[a,b]} \sum_{i=0}^{n} |l_i(x)|$$

Proposition 2.1.10 (Norme de l'opérateur d'interpolation de Lagrange et constante de Lebesgue)

$$|||L_n||| = \Lambda_n$$

Démonstration. On vient de montrer que $|||L_n||| \leqslant \Lambda_n$, montrons que l'inégalité est atteinte pour une certaine fonction g telle que $\|g\| = 1$.
Les l_i sont continues sur $[a,b]$, donc $\sum_{i=0}^{n} |l_i(.)|$ est également continue. On en déduit qu'il existe y dans $[a,b]$ tel que $\Lambda_n = \sum_{i=0}^{n} |l_i(y)|$.
On peut toujours construire une fonction $g : [a,b] \to \mathbb{R}$ telle que

- Pour $i \in \{0,n\}$, $g(x_i) = \text{signe}(l_i(y)) \in \{-1,1\}$
- $\|g\| = 1$.

Pour ce choix de g, on remarque enfin que

$$\|L_n g\| = \sup_{x\in[a,b]} \left| \sum_{i=0}^{n} g(x_i) l_i(x) \right| \geqslant \left| \sum_{i=0}^{n} g(x_i) l_i(y) \right| = \sum_{i=0}^{n} |l_i(y)| = \Lambda_n,$$

d'où

$$\Lambda_n \leq \|L_n g\| \leq |||L_n|||.$$

■

2.1.5 Lien entre stabilité et convergence

Théorème 2.1.11 (Erreur et constante de Lebesque)
Soit $f : [a,b] \to \mathbb{R}$ et $X^n = (x_0, ..., x_n)$. On a

$$\|f - L_n f\| \leqslant E_n^*(f)(1 + \Lambda_n)$$

où

$$E_n^*(f) = \inf_{P\in\mathbb{P}_n} \|f - P\|$$

Démonstration. Soit f une fonction continue sur $[a,b]$. On pose

$$p_n^* = \arg\min_{P\in\mathbb{P}_n} \|f - P\|$$

(L'existence de p_n^* peut se montrer par le fait que $K = \{p \in \mathbb{P}_n : \|f - p\|_{\infty,[a,b]} \leq M\}$ est fermé, borné, donc compact dans $\mathbb{P}_n$).

Alors,

$$\begin{aligned} \|f - L_n f\| &\leqslant \|f - p_n^*\| + \|p_n^* - L_n f\| \\ &\leqslant E_n^*(f) + \|L_n p_n^* - L_n f\| \\ &\leqslant E_n^*(f) + \Lambda_n \|p_n^* - f\| \\ &\leqslant E_n^*(f)(1 + \Lambda_n) \end{aligned}$$

■

2.1.6 Ordre de grandeur de la constantes de contrôle de l'erreur pour n grand

Proposition 2.1.12
Dans le cas de $n+1$ points d'évaluations équidistants sur $[a,b]$, i.e. $x_i = a + i\frac{b-a}{n}$ pour $i \in [\![0;n]\!]$, on a les équivalents suivants

$$\Lambda_n^{\text{equi}} \sim \frac{2^{n+1}}{e.n.\ln(n)}$$

$$\left\|\omega_{n+1}^{\text{equi}}\right\| \sim 2\left(\frac{b-a}{e}\right)^{n+1}$$

2.2 Interpolation aux noeuds de Tchebychev

Dans cette section une alternative aux noeuds équidistants, pour laquelle les constantes de contrôle de l'erreur Λ_n et $\|\omega\|_n$ sont meilleures. Plus précisément, on va choisir les noeuds d'interpolation (sur l'intervalle $[-1,1]$ comme étant les racines du $n+1^{eme}$ polynôme de Tchebychev. Les propriétés de bases de ces polynômes sont rappelées ci-dessous.

2.2.1 Définition et premières propriétés

Définition 2.2.1 (Polynômes de Tchebychev)**.** *On définit les polynômes de Tchebychev pour* $x \in [-1,1]$

$$t_n : x \mapsto \cos(n \arccos x)$$

Proposition 2.2.2
Les (t_n) *sont des polynômes définis par la relation de récurrence*

$$\begin{cases} t_0(x) = 1 \\ t_1(x) = x \\ t_{n+1}(x) + t_{n-1}(x) = 2xt_n(x), n \geqslant 1 \end{cases}$$

Démonstration. Si le système précédent est vrai, la relation de récurrence montre que les (t_n) sont des polynômes. Montrons que le système est vrai :
On pose $\theta = \arccos x$ pour x dans $[-1,1]$. On a par trigonométrie

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

■

Corollaire 2.2.3
t_n *est de degré* n *et de coefficient dominant* 2^{n-1}.

Proposition 2.2.4
t_n *a* n *racines distinctes dans* $[-1,1]$

$$\forall k \in [\![0; n-1]\!], x_k = \cos\left(\frac{(k+\frac{1}{2})\pi}{n}\right)$$

Démonstration. Soit x dans $[-1,1]$. On dispose d'un unique θ dans $[0,\pi]$ tel que $x = \cos(\theta)$.

$$\begin{aligned} t_n(x) = 0 &\Leftrightarrow \cos(n\theta) = 0 \\ &\Leftrightarrow n\theta \equiv \frac{\pi}{2}[\pi] \\ &\Leftrightarrow \theta \equiv \frac{\pi}{2n}\left[\frac{\pi}{n}\right] \end{aligned}$$

d'où le résultat. ■

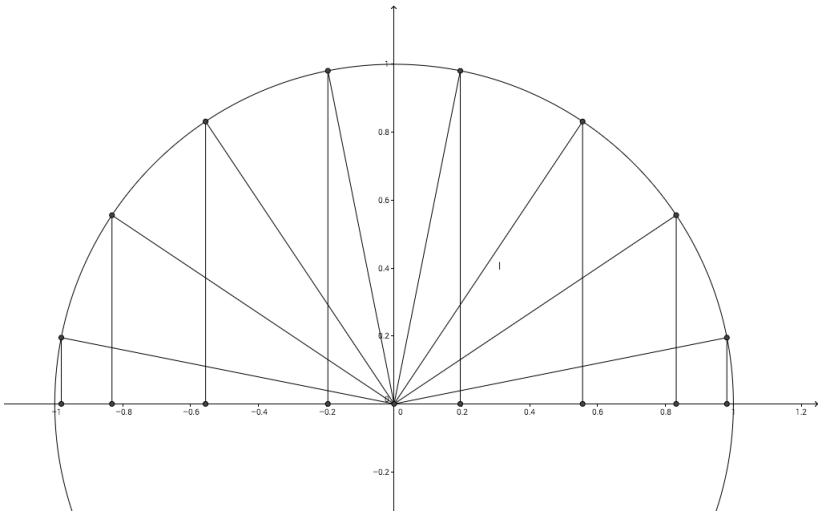


FIGURE 2.2 – Noeuds d'évaluation de Tchebychev pour $n = 8$

Définition 2.2.5 (Noeuds de Tchebychev)**.** *Les points d'interpolation de Tchebychev d'ordre* $n-1$ *sont les* n *racines de* t_n *:* $\{x_0, ..., x_{n-1}\}$*. Il s'agit de l'ordre* $n-1$ *car le polynôme de Lagrange est de degré* $n-1$ *pour* n *points d'interpolations.*

Remarque 2.2.6. *Étant donné que les* (x_i) *sont les racines de* t_n*,et que l'on connaît le coefficient dominant de* t_n*, on peut écrire*

$$t_n = 2^{n-1}\omega_n$$

2.2.2 Interpolation de Tchebychev sur $[a, b]$

Pour se ramener à [-1,1], on considère le changement de variable affine

$$\begin{aligned} \varphi_{[a,b]} : [-1,1] &\to [a,b] \\ u &\mapsto x = \frac{a+b}{2} + \left(\frac{b-a}{2}\right) u \end{aligned} \tag{2.1}$$

On vérifie que $\varphi_{[a,b]}$ est bien un transformation affine entre $[-1, 1]$ et $[a, b]$.

On note (u_k) les racines de t_n sur $[-1, 1]$ et on pose pour k dans $[\![0; n]\!]$

$$x_k = \varphi_{[a,b]}(u_k)$$

On note par la suite $\omega_{n,T}^{[a,b]}$ le polynôme nodal de Tchebychev pour n points d'interpolation sur $[a, b]$. Rappelons que la norme du polynôme nodal est cruciale pour contrôler l'erreur d'interpolation (*c.f.* Corollaire 2.1.8). Contrairement au cas des noeuds équidistants, on peut calculer de manière simple et explicite la valeur de $\|\omega\|_n$ aux noeuds de Tchebychev :

Proposition 2.2.7
La norme du polynôme nodal aux n *noeuds de Tchebychev est donnée par*

$$\left\|\omega_{n,T}^{[a,b]}\right\| = 2\left(\frac{b-a}{4}\right)^n$$

Démonstration. Soit u dans $[-1,1]$

$$\begin{aligned}\omega_{n,T}^{[a,b]} \circ \varphi_{[a,b]}(u) &= \prod_{k=0}^{n-1} \left(\frac{b-a}{2}\right)(u-u_k) \\ &= \left(\frac{b-a}{2}\right)^n \prod_{k=0}^{n-1}(u-u_k) \\ &= \left(\frac{b-a}{2}\right)^n \omega_{n,T}^{[-1,1]}(u) \\ &= 2\left(\frac{b-a}{4}\right)^n t_n(u)\end{aligned}$$

or, $\|t_n\| = \|\cos(n \arccos)\| = 1$, d'où le résultat. ■

Corollaire 2.2.8
On déduit du théorème précédent et de la proposition 2.1.12 :

$$\frac{\left\|\omega_{n,T}^{[a,b]}\right\|}{\left\|\omega_{n,\text{equi}}^{[a,b]}\right\|} \sim \left(\frac{e}{4}\right)^n \qquad (\ll 1 \text{ lorsque } n \to \infty).$$

La deuxième constante intervenant dans le contrôle de la stabilité et de l'erreur est la constante de Lebesgue Λ_n (*c.f.* Proposition 2.1.10 et Théorème 2.1.11). Le résultat suivant (admis) est à comparer avec l'ordre de grandeur de Λ_n pour des noeuds équidistants donnée par la proposition 2.1.12.

Proposition 2.2.9 (Constante de Lebesgue aux noeuds de Tchebychev)
Un équivalent pour n grand de la constante de Lebesgue aux noeuds e Tchebychev sur $[a,b]$ est donné par

$$\Lambda_n^{\text{Tcheby}} \sim_{n\to\infty} \frac{2}{\pi}\ln(n)$$

Ainsi $\Lambda_n^{\text{Tcheby}} \ll \Lambda_n^{equi} \sim \frac{2^{n+1}}{e.n.\ln(n)}$

2.3 Différences divisées, Base de Newton

On cherche dans cette section un moyen efficace en temps de calcul pour calculer numériquement l'interpolateur de Lagrange de f. En effet, l'évaluation 'naïve' à partir de la définition $O(n^2)$ opérations élémentaires (multiplications par $(x-x_i)$ ou (x_i-x_j)) (exercice). La procédé d'évaluation ci-dessous est en $O(n)$ opérations élémentaires.

2.3.1 Base de Newton

Définition 2.3.1 (Base de Newton)**.** *Soit $X^n = (x_1, ..., x_n)$ dans $[a,b]$. On définit les polynômes suivants*

$$\omega_k = \prod_{i=0}^{k-1}(X - x_i) \in \mathbb{P}_k$$

Remarque 2.3.2. *Il s'agit des polynômes nodaux d'ordre k.*

Proposition 2.3.3
La famille $B = \{\omega_0, ..., \omega_n\}$ est une base de $\mathbb{P}_n$.

But :exprimer $L_n f$ dans cette base, autrement dit trouver des coefficients $a_0, \ldots, a_n$ tels que $L_n f = \sum_{i=0}^{n} a_i \omega_i$.

2.3.2 Différences divisées

Soit $k \in [\![1; n]\!]$. On considère le polynôme de $\mathbb{P}_k$

$$\Delta_k = L_k f - L_{k-1} f$$

On remarque que l'on en connaît k racines puisque

$$\Delta_k(x_0) = ... = \Delta_k(x_{k-1}) = 0.$$

On en déduit que, pour tout $k \in [\![1; n]\!]$, il existe $a_k \in \mathbb{R}$ tel que

$$\Delta_k(X) = a_k \omega_k(X).$$

Ainsi

$$\forall k \in [\![1; n]\!], L_k f = a_k \omega_k + L_{k-1} f,$$

Ainsi, il existe bien $(a_0, \ldots, a_n)$ tels que

$$L_n f = \sum_{j=0}^{n} a_j \omega_j, \text{ et } \forall k \leq n, L_k f = \sum_{j=0}^{k} a_j \omega_j \tag{2.2}$$

Intérêt : si on est capable de calculer les a_j pour $j \leq n-1$ (pour n noeuds), et si on rajoute un $n+1^{eme}$ noeud, on n'a plus qu'à calculer le dernier coefficient a_{n+1} (on ne doit pas tout recalculer).

Définition 2.3.4 (Différences divisées)**.** *Les (a_k) sont appelés différences divisées de f. On note*

$$a_k = f_{[x_0,...,x_k]}$$

Remarque 2.3.5. *On a l'égalité suivante*

$$a_0 = f_{[x_0]} = f(x_0)$$

Proposition 2.3.6
On peut calculer les coefficients récursivement

$$f_{[x_0,...,x_k]} = \frac{f_{[x_0,...,x_{k-1}]} - f_{[x_1,...,x_k]}}{x_0 - x_k}$$

et plus généralement

$$\forall i < j, f_{[x_i,...,x_j]} = \frac{f_{[x_i,...,x_{j-1}]} - f_{[x_{i+1},...,x_j]}}{x_i - x_j} \tag{2.3}$$

Démonstration. exercice de TD ■

Intérêt de cette proposition : suggère une méthode itérative pour calculer les $f_{[x_0,...,x_k]}$, à partir d'une table triangulaire.

2.3.3 Méthode de Horner pour l'évaluation

c.f. TD : La méthode de Horner permet d'évaluer $L_n f(x)$ en $O(n)$ opérations, étant données les différences divisées $f_{[x_0,...,x_k]}, k = 0, \ldots, n$.

Chapitre 3

Intégration numérique : méthodes de quadrature

3.1 Introduction : méthodes de quadrature, méthodes de Newton-Cotes

But : Approcher des intégrales de type

$$I(f) = \int_a^b f(x)\,\mathrm{d}x \qquad (a < b \in \mathbb{R})$$

Pratique courante : changement de variable pour se ramener à une intégration sur $[-1, 1]$: $m = (a+b)/2$, $h = (b-a)/2$, $g(u) = f(m + hu)$, $u \in [-1, 1]$.

$$I(f) = \frac{h}{2} \int_{-1}^{1} g(u)\,\mathrm{d}u$$

Les méthodes de quadratures définies ci-dessous sont des opérateurs $\widehat{I} : f \mapsto \widehat{I}(f)$ destinés à approcher $I(f)$.

Définition 3.1.1 (Méthodes de quadratures (MQ))**.**

1. *Une MQ* simple*(MQS) sur $[a, b]$ est un opérateur*

$$f \mapsto \widehat{I}(f) = (b-a) \sum_{i=0}^{n} \lambda_i f(x_i), \tag{3.1}$$

   avec $\lambda_i > 0$, $\sum_{i=0}^{n} \lambda_i = 1$, $x_i \in [a, b]$, $\quad i = 0, \ldots, n$.

2. *Soit $\alpha_0 = a < \alpha_1 < \ldots \alpha_M = b$ une subdivision de $[a, b]$. Une MQ Composite (MQC) est un opérateur de type*

$$f \mapsto \widehat{I}(f) = \sum_{m=0}^{M-1} (\alpha_{m+1} - \alpha_m) \sum_{i=0}^{n_m} \lambda_{m,i} f(x_{m,i}), \tag{3.2}$$

   avec pour tout $m \in \{0, \ldots, M-1\}$:
   $\lambda_{m,i} > 0$, $\sum_{i=0}^{n_m} \lambda_{m,i} = 1$, $x_{m,i} \in [\alpha_m, \alpha_{m+1}]$, $\quad i = 0, \ldots, n_m$.

De manière générale on étudie les propriétés des MQS et l'on en déduit celles de MQC.

Lien avec l'interpolation de Lagrange, rang d'une MQ Les MQS que nous verrons seront toujours de type

$$\widehat{I}(f) = I(L_n f)$$

où $L_n f$ est le polynôme d'interpolation de Lagrange aux noeuds $x_0, \dots, x_n$. Une telle méthode est dite **de rang** n. Le membre de droite est bien de type (3.1). En effet

$$\begin{aligned} I(L_n f) &= \int_a^b \sum_{i=0}^n f(x_i) l_i(x)\,\mathrm{d}x \qquad (l_i :\ i^{eme} \text{ polynôme de Lagrange}) \\ &= \sum_{i=0}^n f(x_i) \underbrace{\int_a^b l_i(x)\,\mathrm{d}x}_{\tilde{\lambda}_i} \\ &= (b-a) \sum_{i=0}^n \lambda_i f(x_i)\ , \end{aligned}$$

avec $\lambda_i = \tilde{\lambda}_i/(b-a) > 0$. Reste à vérifier que $\sum_i \tilde{\lambda}_i = (b-a)$. Or

$$\sum_i \tilde{\lambda}_i = \int_a^b \sum_{i=0}^n l_i(x)\,\mathrm{d}x,$$

Le polynôme $\sum_{i=0}^n l_i$ est de degré au plus n et vaut 1 en $n+1$ points. C'est donc le polynôme constant égal à 1, d'où le résultat.

Cas particulier : méthodes de Newton-Cotes C'est le cas particulier où les points d'interpolation sont équidistants. L'avantage principal est celui de la simplicité d'implémentation

Définition 3.1.2 (Méthode de Newton-Cotes de rang n (NC_n))**.** *LA MQS de Newton-Cotes* ***de rang*** *n sur $[a,b]$ est la méthode données par $\widehat{I}(f) = I(L_n f)$ où $L_n f$ est le polynôme d'interpolation de Lagrange de f aux noeuds* ***équidistants***

$$(x_0 = a, \dots, x_i = a + i\frac{b-a}{n}, \dots, x_n = b).$$

Contrôle de l'erreur : ordre d'une MQ

Définition 3.1.3 (Ordre d'une MQ)**.** *Une MQS (*resp. *une MQC) est dite* ***d'ordre*** *N si elle est*

1. *Exacte pour tout les polynômes de degré $\leq N$,* i.e., *si $\widehat{I}(p) = I(p)$ pour tout $p \in \mathbb{P}_N$. (*resp. *pour toute fonction polynomiale par morceaux sur la subdivision $(\alpha_m)_{m=0:M}$)), et*
2. *Inexacte pour au moins un polynôme $p \in \mathbb{P}_{N+1}$ (*resp. *une fonction polynomiale par morceaux de degré $n+1$).*

Remarque 3.1.4 (Rang et ordre)**.** *Attention à ne pas confondre le rang d'une méthode de quadrature(= le degré n du polynôme de Lagrange sur lequel est basée l'approximation) et son ordre (le plus grand entier N pour lequel la méthode est exacte pour les polynômes de degré N. On verra plus loin que de manière générale $N \geq n$ mais l'inégalité peut être stricte.*

Par linéarité de $\hat{I}$ et de I, et par changement de variable, on a le résultat suivant : notons $X^0 = \mathbf{1}, X, X^2, \ldots, X^N, \ldots$ les polynômes de la base canonique de $\mathbb{P}_N$. Une MQS $\widehat{I}$ est d'ordre N sur $[a, b]$ si et seulement si

1. $\widehat{I}(X^i) = I(X^i)$ sur $(-1, 1]$ pour $i = 1, \ldots, N$ et
2. $\widehat{I}(X^{N+1}) \neq I(X^{N+1})$ sur $[-1, 1]$.

3.2 Exemples

3.2.1 Rectangles à gauche / à droite

1. Rectangles à Gauche :
   - MQS : $\widehat{I}(f) = (b-a)f(a)$
   - MQC : $\widehat{I}(f) = \sum_{m=0}^{M}(\alpha_{m+1} - \alpha_m)f(\alpha_m)$.
2. Rectangles à droite : Idem en remplaçant $f(a)$ par $f(b)$ et $f(\alpha_m)$ par $f(\alpha_{m+1})$.

Ordre de la MQS des rectangles (à gauche ou à droite)
On vérifie pour $[a, b] = [-1, 1]$, que $\widehat{I}(\mathbf{1}) = I(\mathbf{1}) = b - a$, mais que $\widehat{I}(X) \neq I(X)$. La méthode est donc d'ordre 0.

3.2.2 Méthode du point Milieu

On pose $x_0 = \frac{a+b}{2}$ et on prend x_0 comme unique point d'interpolation. On obtient

$$\widehat{I}(f) = (b-a)f(\frac{a+b}{2}).$$

On montre comme au paragraphe précédent que la méthode est d'ordre 1.

3.2.3 Méthode des trapèzes

C'est le nom donné à la méthode de Newton-Cotes de rang 1 (NC_1). Autrement dit on prend deux points d'interpolation $x_0 = a, x_1 = b$, de sorte que

$$L_1 f(x) = f(a)\frac{x-b}{a-b} + f(b)\frac{x-a}{b-a} = \frac{(x-a)f(b) + (b-x)f(a)}{b-a}$$

($L_1 f$ est la la droite passant par $(a, f(a))$ et $(b, f(b))$). Ainsi

$$\widehat{I}(f) = \int_a^b L_1 f(x)\,\mathrm{d}x = \ \ldots(\text{calcul})\ldots = (b-a)\frac{f(a)+f(b)}{2}$$

(c'est l'aire du trapèze sous le segment de droite entre les deux points d'interpolation).

Une formule simple pour la méthode composite associée
c.f. TP4.

Ordre de la méthode des trapèzes On vérifie que la méthode est d'ordre 1.

3.2.4 Méthode de Cavalieri-Simpson

C'est le nom donné à la méthode de Newton-Cotes de rang 2 (NC_2) : on intègre le polynôme de Lagrange aux points $x_0 = a, x_1 = (a+b)/2,\ x_2 = b$. On montre que les poids de la méthode sont

$$\lambda_0 = \frac{1}{6}\,,\ \lambda_1 = \frac{2}{3}\,,\ \lambda_2 = \frac{1}{6}\,.$$

(exercice : vérifiez-le par intégration des polynômes de Lagrange $l_i, i = 0, 1, 2$).

3.3 Ordre des méthodes de Newton-Cotes

On généralise les calculs d'ordre des méthodes données en exemple à la section précédente. Soit $\widehat{I}$ une méthode NC_n. Par définition $\widehat{I}(f) = I(L_n f)$ avec $L_n f$ le polynôme de Lagrange interpolé en $n+1$ points équidistants. Lorsque f est un polynôme de degré $\leq n$, on a $L_n f = f$ (l(interpolation est exacte), d'où $\hat{I}(f) = I(f)$, la méthode NC_n est exacte. l'ordre d'une méthode NC_n est donc toujours supérieur ou égal à n. Lorsque n est pair, on gagne un ordre de précision, comme résumé dans la proposition ci-dessous.

Proposition 3.3.1
Pour tout $n \in \mathbb{N}$, l'ordre N de la méthode NC_n est

- $N = n$ *si* n *est impair*
- $N = n+1$ *si* n *est pair.*

Démonstration. on a montré que l'ordre est $\geq n$.

- Pour n impair, on admet qu'il y a égalité.
- Pour n pair, on va montrer que l'ordre est supérieur ou égal à $n+1$, et on admettra l'égalité.

Soit $n \in \mathbb{N}$, pair. Par linéarité / changement de variable, il suffit de montrer que sur l'intervalle $[-1, 1]$, $\widehat{I}(f) = I(f)$ avec $f(x) = x^{n+1}$. Par imparité, $I(f) = 0$. D'autre part, par définition de $\widehat{I}(f)$, on a

$$\widehat{I}(f) = \sum_{i=0}^{n} x_i^{n+1} \int_{-1}^{1} l_i(x)\,\mathrm{d}x.$$

De plus, on a, par symétrie des x_i et des l_i,

$$x_{n/2} = 0\,,\ \text{et } \forall i \in 1, \ldots \frac{n}{2} - 1,\ x_{n-i} = -x_i.$$

enfin pour tout $i \in \{0, \ldots, n\}$, $l_{n-i}(x) = l_i(-x) \quad (\forall x \in [-1, 1])$, d'où

$$\int l_i = \int l_{n-i}.$$

Ainsi

$$\begin{aligned}\widehat{I}(f) &= \sum_{i=0}^{\frac{n}{2}-1} x_i^{n+1} \int l_i + (-x_i)^{n+1} \int l_i \\ &= 0\end{aligned}$$

Ainsi $I(f) - \widehat{I}(f) = 0$: la méthode est donc bien d'ordre $\geq n+1$. ■

3.4 Stabilité

A cause des erreurs de mesure/ d'arrondi, on n'a jamais exactement accès aux valeurs $f(x_i)$ mais plutôt à des valeurs perturbées $\tilde{f}(x_i) = f(x_i) + \delta f(x_i)$, où δf est la 'perturbation' de la fonction f. Par linéarité de l'opérateur $\widehat{I}$, ceci induit une perturbation du résultat de la méthode :

$$\widehat{I}(\tilde{f}) - \widehat{I}(f) = \widehat{I}(\tilde{f} - f) = \widehat{I}(\delta f).$$

On pose $g = \delta f$. On considère des fonctions (et des perturbations) bornées sur $[a, b]$ et on utilise la norme infinie sur $[a, b]$, $\|f\| = \|f\|_{\infty,[a,b]}$. L'analyse de la stabilité (cf. Chapitre 1) consiste alors à trouver une constante K (si elle existe) telle que pour toute fonction g bornée,

$$\left|\widehat{I}(g)\right| \leq K\|g\|.$$

Plus précisément on cherche à donner la 'meilleure' constante K telle que la condition ci-dessus soit vérifiée, c'est-à-dire le conditionnement absolu $K_{abs} = \sup_{\|g\|=1} |\widehat{I}(g)|$

3.4.1 Stabilité d'une MQS

Par définition des méthodes de quadratures simples sur $[a, b]$, on peut écrire

$$\begin{aligned}
\left|\widehat{I}(g)\right| &= \left|\sum_{i=0}^{n} g(x_i) \int_a^b l_i(x)\,\mathrm{d}x\right| \\
&\leq \|g\|_{\infty,[a,b]} \int_a^b \underbrace{\sum_{0}^{n} |l_i(x)|}_{\leq \Lambda_n = \sup_{u\in[a,b]} \sum |l_i(u)|} \mathrm{d}x \\
&\leq \|g\|_{\infty,[a,b]} \Lambda_n (b-a),
\end{aligned}$$

où Λ_n est la constante de Lebesgue (norme de l'opérateur d'interpolation de lagrange, *c.f.* Définition 2.1.9). La méthode est donc stable et le conditionnement absolu vérifie $K_{abs} \leq \Lambda_n(b-a)$. On peut montrer (comme pour le calcul du conditionnement de l'interpolateur de Lagrange, Proposition 2.1.10) qu'il y a égalité. On retiendra

Proposition 3.4.1
Le conditionnement absolu d'une MQS sur $[a; b]$ basée sur l'interpolation aux noeuds $(x_0, \ldots, x_n)$ est

$$K_{abs} = \Lambda_n(b-a)$$

où Λ_n est la constante de Lebesgue pour les noeuds $(x_0, \ldots, x_n)$.

La constante Λ_n croissant rapidement avec n, on voit que le conditionnement se détériore pour des MQS de rang n élevé.

3.4.2 Stabilité d'une MQC

Soit $\widehat{I}_M$ Une méthode composite avec M sous intervalles $\alpha_m, \alpha_{m+1}, m = 0, \ldots, M-1$, sur chacun desquels on applique une méthode simple de rang n notée $\hat{I}_{[\alpha,m,\alpha_{m+1}]}$. Par définition

on a $\widehat{I}_M = \sum_{i=0}^{M-1} \widehat{I}_{[\alpha,m,\alpha_{m+1}}$. On en déduit la majoration suivante, pour toute fonction g bornée sur $[a,b]$:

$$\begin{aligned}
|\widehat{I}_M(g)| &= \Big| \sum_{i=0}^{M-1} \widehat{I}_{[\alpha,m,\alpha_{m+1}]}(g) \Big| \\
&\leq \sum_{i=0}^{M-1} \Big|\widehat{I}_{[\alpha,m,\alpha_{m+1}]}(g)\Big| \\
&\leq \sum_{i=0}^{M-1} \Lambda_n(\alpha_m - \alpha_{m+1})\|g\|_{\infty,[\alpha_m,\alpha_{m+1}]}\Big| \\
&\leq \lambda_n \|g\|_\infty (b-a).
\end{aligned}$$

Comme au paragraphe précédent, il est facile de construire une fonction g telle qu'il y ait égalité. On obtient ainsi le même résultat que pour une MQS.

Proposition 3.4.2
Le conditionnement absolu d'une MQC sur $[a;b]$ construite avec M sous-intervalles et une méthode simple de rang n sur chaque sous-intervalle est le même que le conditionnement d'une MQS de rang n, c'est-à-dire

$$K_{abs} = \Lambda_n(b-a)$$

où Λ_n est la constante de Lebesgue de la MQS de rang n choisie.

Conclusion : il est beaucoup plus intéressant d'utiliser des MQC avec un grand nombre M de sous-intervalles et un rang n modéré, plutôt qu'une méthode simple de rang $O(nM)$, car la stabilité ne dépend que du rang de la méthode simple et est meilleure pour des rangs faibles.

3.5 Majorations de l'erreur

3.5.1 Erreur d'une MQS d'ordre N

On définit l'erreur d'une méthode de quadrature simple d'ordre N (donc de rang $n \leq N$) appliquée à une fonction f sur l'intervalle $[a,b]$, par

$$EI_N^{(a,b)}(f) := \widehat{I}(f) - I(f).$$

Soit $\widehat{I}$ une méthode d'ordre N. On suppose que $f \in \mathcal{C}^{(N+1)}[a,b]$. La formule de Taylor avec reste exacte donne, pour $x \in [a,b]$,

$$f(x) = p_N(x) + \underbrace{\frac{f^{(N+1)}(\theta_x)}{(N+1)!}(x-a)^{N+1}}_{g_N(x)}, \tag{3.3}$$

où $\theta_x \in [a,x]$ et $p_N \in \mathbb{P}_N$. Comme I et $\widehat{I}$ sont des opérateurs linéaires, l'erreur EI l'est aussi. Ainsi

$$EI_N^{(a,b)}(f) = EI(p_N) + EI(g_N) = EI(g_N)\,,$$

puisque $\widehat{I}$ est exacte sur $\mathbb{P}_N$. On obtient la majoration suivante de l'erreur

$$\left|EI_N^{(a,b)}(f)\right| = \left|I(g_N) - \widehat{I}(g_N)\right| \leq |I(g_N)| + \left|\widehat{I}_N(g_N)\right| \tag{3.4}$$

D'après (3.3),

$$|I(g_N)| \leq \frac{\|f^{(N+1)}\|}{(N+1)!}\frac{(b-a)^{N+2}}{N+2} \;=\; \frac{\|f^{(N+1)}\|}{(N+2)!}(b-a)^{N+2}.$$

D'autre part

$$\begin{aligned}\left|\widehat{I}(g_N)\right| &= \left|(b-a)\sum_0^n g_N(x_i)\lambda_i\right| \\ &\leq (b-a)\|g_N\| \\ &\leq \frac{\|f^{(N+1)}\|(b-a)^{N+1}}{(N+1)!}(b-a) \\ &\leq \frac{\|f^{(N+1)}\|}{(N+1)!}(b-a)^{N+2}\end{aligned}$$

Au vu de (3.4) et des deux dernières majorations, on a finalement

$$[EI_N^{(a,b)}(f)| \leq C_N\|f^{(N+1)}\|(b-a)^{N+1}, \tag{3.5}$$

avec $C_N = 1/(N+1)! + 1/(N+2)!$.

Cette borne ne permet pas d'assurer que l'erreur tende vers 0 si la fonction n'est pas très 'régulière', au sens où la norme des dérivées d'ordre élevé 'explose', ou si $(b-a) > 1$. Ceci suggère encore une fois d'utiliser des méthodes composites avec de petits sous-intervalles et un ordre modéré.

3.5.2 Erreur d'une MQC d'ordre N sur des intervalles de même longueur

Soit $EI_{N,M}^{(a,b)}(f)$ l'erreur d'une méthode composite sur M sous-intervalles $[\alpha_m, \alpha_{m+1}]$ sur lesquels on applique une méthode d'ordre N, $\widehat{I}_{[\alpha_m,\alpha_{m+1}]}$. On considère des α_i équidistants $\alpha_m = a + m\frac{b-a}{M}$ et on note $h = \frac{b-a}{M}$.

N.B. On ne suppose pas que la MQS utilisée sur chaque sous-intervalle utilise des noeuds équirépartis.

L'erreur s'écrit alors

$$\begin{aligned}EI_{N,M}^{(a,b)}(f) &= \sum_{m=0}^{M-1}\widehat{I}_{[\alpha_m,\alpha_{m+1}]} - \int_a^b f \\ &= \sum_{m=0}^{M-1}\widehat{I}_{[\alpha_m,\alpha_{m+1}]} - \int_{\alpha_m}^{\alpha_{m+1}} f \\ &= \sum_{m=0}^{M-1} EI_n^{(\alpha_m,\alpha_{m+1}}(f)\end{aligned}$$

D'où

$$\begin{aligned}\left|EI_{N,M}^{(a,b)}(f)\right| &\leq \sum_{m=0}^{M-1} EI_n^{(\alpha_m,\alpha_{m+1}}(f) \\ &\leq \sum_{m=0}^{M-1} C_N \|f^{(N+1)}\|_{\infty,[\alpha_m,\alpha_{m+1}]}\ (\alpha_{m+1}-\alpha_m)^{N+2} \\ &\leq C_N \|f^{(N+1)}\|_{\infty,[a,b]}\ M\Big(\underbrace{\frac{b-a}{M}}_{h}\Big)^{N+2} \\ &=\leq C_N \|f^{(N+1)}\|_{\infty,[a,b]}(b-a)\ h^{N+1} \qquad (3.6)\end{aligned}$$

On gagne donc un facteur $\Big[h/(b-a)\Big]^{N+1} = M^{-(N+1)}$ par rapport à la borne d'erreur (3.5) pour une MQS. En utilisant une MQS d'ordre N modéré (donc de faible rang et ainsi avec une bonne stabilité), il est toujours possible d'augmenter M (diminuer h), et d'avoir un bon contrôle de l'erreur.

3.5.3 Erreur des méthodes de Newton-Cotes

On conclut cette partie sur un résultat (admis) sur l'erreur des méthodes de Newton-Cotes (*i.e.*les MQS avec noeuds équidistants et les MQC associées), pour lesquelles les inégalités (3.5) et (3.6) sont des égalités.

Proposition 3.5.1
Il existe une suite de constantes $(K_N)_{n\in\mathbb{N}} > 0$ indépendantes de l'intervalle d'intégration (a,b) et de la fonction f à intégrer telles que pour tout $n \in \mathbb{N}$, pour toute fonction $f \in \mathcal{C}^{(N+1)}[a,b]$:

1. *L'erreur de la MQS Newton Cotes d'ordre N, est*

$$EI_N^{(a,b)}(f) = K_N f^{(N+1)}(\theta_N)\ (b-a)^{N+2},$$

   où $\theta_N \in [a,b]$,

2. *La MQC de Newton Cotes d'ordre N et de pas $h = \frac{b-a}{M}$, il existe $\theta_{N,M} \in [a,b]$, tel que*

$$EI_{N,M}^{(a,b)}(f) = K_N f^{(N+1)}(\theta_{N,M})\ (b-a)\ h^{N+1}$$

3.6 Évaluations de l'erreur a posteriori

L'inconvénient de l'analyse de l'erreur dans la section précédente est d'impliquer des quantités le plus souvent inconnues (les dérivées de f, évaluées en des points θ_N inconnus) Une autre manière dite 'a posteriori' d'évaluation de l'erreur des méthodes composites repose sur la comparaison entre les résultats de la méthode de pas $h = (b-a)/M$ et celle de pas $h/2 = (b-a)/(2M)$. En effet la proposition 3.5.1 2. implique

$$\begin{aligned}\frac{EI_{N,M}(f)}{EI_{N,2M}(f)} &= \frac{f^{(N+1)}(\theta_{N,M})}{f^{(N+1)}(\theta_{N,2M})}\frac{h^{N+1}}{(h/2)^{N+1}} \\ &= \frac{f^{(N+1)}(\theta_{N,M})}{f^{(N+1)}(\theta_{N,2M})}\ 2^{N+1} \\ &\approx 2^{N+1}\end{aligned}$$

En supposant que $f^{(n+1)}(\theta_{N,2M}) \approx f^{(n+1)}(\theta_{N,M})$. En notant I_M (*resp.* I_{2M}) la MQC avec M (*resp.* $2M$) sous-intervalles, la dernière expression donne une relation simple entre I, I_M et I_{2M}. Une inversion simple permet d'exprimer l'erreur $I_{2M} - I$ en fonction de la différence $I_{2M} - I_M$.
Exemple : Analyse a posteriori pour la méthode de Cavalieri Simpson (NC_2, ordre $N = 3$) : *c.f.* TP.

3.7 Extrapolation de Richardson, application aux trapèzes (méthode de Romberg)

3.7.1 Principe de la méthode de Richardson

L'extrapolation de Richardson est une méthode générale pour estimer efficacement la valeur en 0 d'une fonction $A : \mathbb{R}^+ \to \mathbb{R}$, à partir d'évaluations en certains points bien choisis $t_i > 0$, $i = 0, \ldots, n$. Dans le cas de l'intégration numérique, on prend $A(h) = \widehat{I}(h)$ où $\widehat{I}$ est une MQC de pas $h = (b-a)/M$. On cherche alors à évaluer $\widehat{I}(0) := \lim_{h\to 0} \widehat{I}(h) = I$ (la vraie intégrale), où la dernière égalité vient du fait que l'erreur tend vers 0 avec h, d'après la majoration (3.6) de l'erreur.

Les points t_i sont choisis de la manière suivante (pour des raisons qui apparaîtront plus loin) : on fixe $t > 0$ et on choisit $\delta \in]0, 1[$, puis on pose

$$t_i = \delta^i t, \quad i = 0, \ldots, n$$

Un estimateur 'naïf' est $\widehat{A}_{naif}(0) = A(\delta^n t)$. Un développement de Taylor d'ordre 1 donne $A(0) = A(\delta^n t) + O(\delta^n t)$. L'erreur est en $O(\delta^n t)$.

L'approximation de Richardson permet de diminuer l'ordre de grandeur de l'erreur en interpolant un polynôme de Lagrange aux noeuds $x_i = t_i = \delta^i t$. Soit $L_n^t A$ le polynôme de Lagrange obtenu. On prend alors comme estimateur, la valeur de l'interpolateur en 0, soit

$$\widehat{A}_{rich}(0) = L_n^t A(0).$$

Analyse de l'erreur de la méthode de Richardson
Dans la suite on appelle $A_n(t) = L_n^t A(0)$ l'estimateur de Richardson décrit ci-dessus. La formule de l'erreur de l'interpolation de Lagrange (Théorème 2.1.7) sur l'intervalle $[a, b] = [0, t]$ donne

$$A_n(t) = L_n^t A(0) = A(0) + \frac{A^{(n+1)}(\theta_n)}{(n+1)!}\, \omega_{n+1}(0),$$

où $\theta_n \in [0, t]$ et ω_{n+1} est le polynôme nodal aux noeuds $\delta^i t, i = 0, \ldots, n$. Enfin,

$$\omega_{n+1}(0) = \prod_{i=0}^{n} (0 - \delta^i t) = (-1)^{n+1} \delta^{\sum_0^n i} t^{n+1} = (-1)^{n+1} \delta^{\frac{n(n+1)}{2}} t^{n+1}$$

L'erreur de la méthode de Richardson est donc d'ordre de grandeur

$$A(0) - A_n(t) = O(\delta^{\frac{n(n+1)}{2}} t^{n+1})$$

le rapport entre l'erreur de Richardson et l'erreur de l'estimateur naïf est en $O(\delta^{\frac{n(n-1)}{2}} t^n)$, ce qui rend très attractive la méthode de Richardson pour $\delta < 1$.

Interprétation de la méthode comme une 'élimination' des termes dans le développement de A en 0
Supposons que A admette un développement limité en 0,

$$A(t) = a_0 + a_1 t + a_2 t^2 + \ldots + O(t^{k+1}).$$

avec $a_0 = A(0)$ et $k \geq n$.

Puisque l'estimateur de Richardson s'écrit comme une combinaison linéaire des $A(\delta^i t)$, on a aussi un développement, à n fixé, de type

$$A_n(t) = b_0^{(n)} + b_1^{(n)} t + b_2^{(n)} t^2 + \ldots + O(t^{k+1})$$

Cependant l'argument du paragraphe précédent montre que

$$A_n(t) \underset{t\to 0}{=} A(0) + O(t^{n+1}) = a_0 + O(t^{n+1}).$$

Par unicité on a

$$b_0^{(n)} = a_0 = A(0) \qquad \text{et} \qquad b_1^{(n)} = b_2^{(n)} = \cdots = b_n^{(n)} = 0.$$

Il est donc préférable d'utiliser $A_n(t)$ plutôt que $A(t)$ (où même $A(\delta^n t)$) pour approcher a_0, puisque le premier terme non nul dans le développement de $A_n(t) - A(0)$ est d'ordre t^{n+1} (plutôt que t).

Implémentation : calcul de $A_n(t)$
On peut utiliser une méthode voisine de celle des différences divisées pour calculer rapidement $A_n(0)$, étant donnés les $A(\delta^i t)$, *c.f.* TP.

3.7.2 Méthode de Romberg

C'est l'application de la méthode de Richardson lorsque $A(t)$ est le résultat de la MQC des trapèzes, de pas $h = \sqrt{t}$, en prenant $\delta = 1/4$. (la raison de ce 'changement de variable' et du choix de δ est expliquée plus bas).

Rappelons que la méthode des trapèzes est une Newton-Cotes de rang 1, donc d'ordre 1. L'erreur de la MQC de pas h est en $O(h^2)$ d'après la Proposition 3.5.1. On note $\widehat{I}(h)$ cette méthode. On utilise le résultat plus fort suivant (admis)

Proposition 3.7.1 (Formule d'Euler Maclaurin)
La MQC des trapèzes de pas h sur a, b, appliquée à une fonction $2k+2$ fois dérivable, admet un développement en 0 :

$$\widehat{I}(h) = \int_a^b f + a_1 h^2 + a_2 h^4 + \ldots + a_k h^{2k} + a_{k+1} f^{2k+2}(\theta)$$

où $\theta \in [a, b]$ et où les a_i dépendent des dérivées d'ordre $2i-1$ de f en a et b.

Ainsi, seuls les termes pairs interviennent dans le développement de $\widehat{I}(h)$ en 0. En posant $t = h^2$ et $A(t) = \widehat{I}(\sqrt{t}) = \widehat{I}(h)$, on a donc

$$A(t) = \underbrace{\int_a^b f}_{A(0)} + \sum_{i=1}^{k} a_i t^i + O(t^{k+1}).$$

Appliquer la méthode de Richardson à $A(t)$ plutôt qu'à $I(h)$ permet donc d'éliminer les n premiers termes non nuls du développement, alors qu'on éliminerait seulement les $n/2$ premiers en appliquant Richardson à la fonction $\widehat{I}(h)$.

Concernant le choix de δ : en prenant $\delta = 1/4$, on a $\delta^i t = (h/2^i)^2$ et on doit évaluer successivement $\widehat{I}(h), \widehat{I}(h/2), \ldots, \widehat{I}(h/2^i), \ldots$. Or, il est peu coûteux d'évaluer $\widehat{I}(h/2)$ lorsque l'on connaît la valeur de $\widehat{I}(h)$ (voir TP 2-3).

Chapitre 4

Équations différentielles

4.1 Existence et unicité des solutions

4.1.1 Problème de Cauchy

Le problème de Cauchy, que l'on cherche à résoudre, est une équation sur la fonction $y : I \to \mathbb{R}^m$:

$$\begin{aligned} y(t_0) &= y_0 \\ y'(t) &= f(t, y(t)), \forall t \in I. \end{aligned}$$

Les données du problème sont :

- l'intervalle de définition $I \subset \mathbb{R}$,
- la fonction $f : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$ qui est continue en y et intégrable en t.
- et la condition initiale (t_0, y_0) où $t_0 \in \mathring{I}$.

4.1.2 Exemple : les équations du mouvement

q est la position
$p = mq'$ est m fois la vitesse
$H(q,p) = V(q) + \frac{\|p\|^2}{2m}$ est l'énergie totale
Si il y a conservation de l'énergie :

$$\begin{aligned} q' &= \frac{1}{m}p = \frac{\partial H}{\partial p}(q,p) \\ p' &= -\nabla V(q) = -\frac{\partial H}{\partial q}(q,p) \end{aligned}$$

Avec les notations précédentes, on a $y = \begin{bmatrix} q \\ p \end{bmatrix} \in \mathbb{R}^6$ et $f(t,y) = f(y) = \begin{bmatrix} \frac{\partial H}{\partial p}(y) \\ -\frac{\partial H}{\partial q}(y) \end{bmatrix}$.

Résoudre le problème de Cauchy revient à simuler le mouvement du système en partant des conditions initiales y_0 en t_0.

Remarque : On aurait pu écrire $q'' = -\frac{\nabla V(q)}{m}$. La méthode pour passer d'une équation différentielle d'ordre quelconque à une équation différentielle d'ordre 1 est générale.

4.1.3 Théorèmes d'existence et unicité des solution

Régularité des solutions. $\forall y$ solution, si f est de classe C^k, alors y est de classe C^{k+1}. En particulier, y est dérivable.

Lemme 4.1.1
La fonction $y : I \to \mathbb{R}^m$ est solution du problème de Cauchy si et seulement si

$$\forall t \in I, y(t) = y_0 + \int_{t_0}^{t} f(u, y(u))du \qquad \textit{(équation intégrale)}$$

Démonstration. $\Rightarrow$: On intègre. $\Leftarrow$: On dérive. ■

Outil de preuve. Soit $C(I, \mathbb{R}^m)$ l'ensemble des fonctions continues de l'intervalle I (qui contient t_0) vers l'ensemble $\mathbb{R}^m$. Soit ϕ la fonction de $C(I, \mathbb{R}^m) \to C(I, \mathbb{R}^m)$ définie par

$$\forall z \in C(I, \mathbb{R}^m) : \forall x \in I, \phi[z](x) = y_0 + \int_{t_0}^{x} f(\tau, z(\tau))d\tau \tag{4.1}$$

Lemme 4.1.2
Soit $r > 0$ et soit ϕ définie dans (4.1). Il existe $T > 0$ tel que
si $\|z - y_0\|_T = \sup_{t\in[t_0-T,t_0+T]}\|z(t) - y_0\| \leq r$, alors $\|\phi[z] - y_0\|_T = \sup_{t\in[t_0-T,t_0+T]}\|\phi[z](t) - y_0\| \leq r$.

Démonstration.

$$\|\phi[z](t) - y_0\| = \|\int_{t_0}^{t} f(\tau, z(\tau))d\tau\| \leq |\int_{t_0}^{t} \|f(\tau, z(\tau))\|d\tau|$$

Fixons $T_0 > 0$. Si $\tau \in [t_0 - T_0; t_0 + T_0]$ et $\|z(\tau) - y_0\| \leq r$, alors $(\tau, z(\tau)) \in [t_0 - T_0; t_0 + T_0] \times B(y_0, r)$.
$[t_0 - T_0; t_0 + T_0] \times B(y_0, r)$ est un ensemble compact et f est continue donc $\exists M > 0$ tel que $\|f(\tau, z(\tau))\| \leq M, \forall \tau \in [t_0 - T_0; t_0 + T_0]$.

Ainsi, $\|\phi[z](t) - y_0\| \leq |\int_{t_0}^{t} \|f(\tau, z(\tau))\|d\tau| \leq |t - t_0|M$
et il suffit de prendre $T = \min(T_0, \frac{r}{M})$ pour avoir le résultat. ■

Soit $\mathcal{F} = C([t_0 - T; t_0 + T] \times B(y_0, r))$. La restriction de ϕ à $\mathcal{F}$ est telle que $\phi_{|\mathcal{F}} : \mathcal{F} \to \mathcal{F}$. On va maintenant écrire $\phi = \phi_{|\mathcal{F}}$.

Théorème 4.1.3 (Cauchy-Lipschitz : existence et unicité des solutions locales)
Si $f : I \times \mathbb{R}^m \to \mathbb{R}^m$ est localement lipschitzienne en y alors $\exists T > 0$ telque le problème de Cauchy avec condition initiale (t_0, y_0) admet une unique solution $y : [t_0 - T; t_0 + T] \to \mathbb{R}^m$.

Démonstration. On va démontrer que ϕ est une contraction et utiliser le théorème du point fixe de Picard. Rappelons que l'espace des fonctions continues muni de la norme sup est un espace complet et donc que le théorème du point fixe s'applique.

Soit $(z_1, z_2) \in \mathcal{F} \times \mathcal{F}$ et soit $k > 0$ tel que f est k-lipschitzienne en y sur $\mathcal{F}$. Comme $\mathcal{F}$ est compact et f est localement lipschitzienne, ce k existe forcément.

$$
\begin{aligned}
\|\phi[z_1] - \phi[z_2]\| &= \sup_{t\in[t_0-T;t_0+T]} \|\phi[z_1](t) - \phi[z_2](t)\| && \text{(déf. } \|\phi[z]\|\text{)}\\
&= \sup_{t\in[t_0-T;t_0+T]} \|y_0 + \int_{t_0}^{t} f(\tau, z_1(\tau))d\tau - y_0 - \int_{t_0}^{t} f(\tau, z_2(\tau))d\tau\| && \text{(déf. } \phi\text{)}\\
&\leq \sup_{t\in[t_0-T;t_0+T]} |\int_{t_0}^{t} \|f(\tau, z_1(\tau)) - f(\tau, z_2(\tau))\|d\tau| && (\|\int f\| \leq \int \|f\|)\\
&\leq \sup_{t\in[t_0-T;t_0+T]} |\int_{t_0}^{t} k\|z_1(\tau) - z_2(\tau)\|d\tau| && (f \text{ est } k\text{-lipschitzienne})\\
&\leq \sup_{t\in[t_0-T;t_0+T]} |\int_{t_0}^{t} k\|z_1 - z_2\|d\tau|\\
&= \sup_{t\in[t_0-T;t_0+T]} |t_0 - t|k\|z_1 - z_2\| = Tk\|z_1 - z_2\|
\end{aligned}
$$

Si $T < \frac{1}{k}$, on peut s'arrêter là. En fait, on peut raffiner un peu le résultat. En effet, par un calcul très similaire, on trouve pour $\phi^2 = \phi \circ \phi$:

$$
\begin{aligned}
\|\phi^2[z_1] - \phi^2[z_2]\| &\leq \sup_{t\in[t_0-T;t_0+T]} |\int_{t_0}^{t} k\|\phi[z_1](\tau) - \phi[z_2](\tau)\|d\tau|\\
&\leq \sup_{t\in[t_0-T;t_0+T]} |\int_{t_0}^{t} k \sup_{t\in[t_0;\tau]} \|\phi[z_1] - \phi[z_2]\|d\tau|\\
&\leq \sup_{t\in[t_0-T;t_0+T]} |\int_{t_0}^{t} k \times |\tau - t_0|k\|z_1 - z_2\|d\tau|\\
&= \sup_{t\in[t_0-T;t_0+T]} \frac{(t-t_0)^2}{2}k^2\|z_1 - z_2\| = \frac{T^2k^2}{2}\|z_1 - z_2\|
\end{aligned}
$$

Par récurrence, on prouve que $\forall p \in \mathbb{N}$, $\|\phi^p[z_1] - \phi^p[z_2]\| \leq \frac{T^p k^p}{p!}\|z_1 - z_2\|$. Or $\lim_{p\to+\infty} \frac{T^p k^p}{p!} = 0$, donc $\exists p, \exists \alpha < 1$ tels que $\forall (z_1, z_2) \in \mathcal{F} \times \mathcal{F}$, $\|\phi^p[z_1] - \phi^p[z_2]\| \leq \alpha\|z_1 - z_2\|$. Par le théorème du point fixe de Picard, ϕ a un point fixe, il est unique et la suite $(\phi^n[z])_{n\in\mathbb{N}}$ converge vers ce point fixe. Notons ce point fixe y. D'après le lemme 4.1.1, $y = \phi(y)$ signifie que y est solution du problème de Cauchy. ■

Théorème 4.1.4 (Unicité globale)
Soient y_1 et y_2 deux solutions de l'équation différentielle sur I (leur condition initiale peut être différente), avec f localement lipschitzienne en y. Si y_1 et y_2 coïncident en un point t_1 de I, alors $y_1(t) = y_2(t)$, $\forall t \in I$.

Démonstration. On considère le premier instant auquel y_1 et y_2 ne coïncident plus :

$$\tilde{t}_0 = \inf\{t \in I, t \geq t_1, y_1(t) \neq y_2(t)\}.$$

D'après le théorème d'unicité locale, il existe $\tilde{T}$ tel que le problème de Cauchy avec conditions initiales $(\tilde{t}_0, y_1(\tilde{t}_0))$ a une unique solution sur $[\tilde{t}_0 - \tilde{T}; \tilde{t}_0 + \tilde{T}]$. Ainsi $y_1(t) = y_2(t)$ sur $[\tilde{t}_0 - \tilde{T}; \tilde{t}_0 + \tilde{T}]$. Cela contredit la définition de $\tilde{t}_0$ donc y_1 et y_2 coïncident sur I tout entier. ■

Remarque : ce théorème ne garantit pas l'existence de solutions sur I, juste l'unicité.

4.2 Équations différentielles linéaires

Définition 4.2.1. *Une équation différentielle $y'(t) = f(t, y(t))$ est* linéaire *si f est affine en y, c'est-à-dire si il existe pour chaque $t \in \mathbb{R}$ une matrice carrée $A(t) \in \mathcal{M}_m(\mathcal{C})$ et un vecteur $b(t) \in \mathbb{C}$ tels que $f(t, y) = A(t)y + b(t)$.*

f est continue dès que A et b sont des fonctions continues.

Pour tout $t \in \mathbb{R}$, f est lipschitzienne en y de rapport $k(t) = \|A(t)\|$ (norme d'opérateur).

Conséquence : Le théorème de Cauchy-Lipschitz s'applique et donc le problème de Cauchy a une unique solution.

4.2.1 Géométrie de l'ensemble des solutions

Proposition 4.2.2
Soit S_0 l'ensemble des solutions sur $\mathbb{R}$ de l'équation différentielle linéaire sans second membre $(E_0) : y'(t) = A(t)y(t)$.

S_0 est un espace vectoriel de dimension m.

Démonstration. Soient x et y deux solutions de (E_0) et soient $\lambda, \mu \in \mathbb{C}$.

$$(\lambda x + \mu y)'(t) = \lambda x'(t) + \mu y'(t) = \lambda A(t)x(t) + \mu A(t)y(t) = A(t)(\lambda x(t) + \mu y(t))$$

donc $\lambda x + \mu y$ est une solution de (E_0) et S_0 est bien un espace vectoriel.

De plus, l'application ϕ_{t_0} d'évaluation au temps t_0 définie par $\phi_{t_0} : \begin{matrix} S_0 \to \mathbb{C}^m \\ y \mapsto y(t_0) \end{matrix}$ est un isomorphisme linéaire d'après le théorème d'existence (surjectivité) et d'unicité (injectivité). Cela implique que $\dim(S_0) = \dim(\mathbb{C}^m) = m$. ■

Proposition 4.2.3
Soit S l'ensemble des solutions sur $\mathbb{R}$ de l'équation différentielle linéaire $(E) : y'(t) = A(t)y(t) + b(t)$ et soit $y_{(1)}$ une solution quelconque de (E).

Alors $S = \{y \in C(\mathbb{R}, \mathbb{C}^m) \ : \ \exists z \in S_0 \ , \ y = y_{(1)} + z\} = y_{(1)} + S_0$ (espace affine).

Démonstration. Soit $y \in S$ et soit $z = y - y_{(1)}$. On a que $z'(t) = y'(t) - y'_{(1)}(t) = A(t)z(t)$ donc $z \in S_0$. On en déduit que $S \subset y_{(1)} + S_0$.

Soit $y \in y_{(1)} + S_0 : y = y_{(1)} + z$ où $z \in S_0$.

$$y'(t) = y'_{(1)}(t) + z'(t) = A(t)y_{(1)}(t) + b(t) + A(t)z(t) = A(t)y(t) + b(t)$$

et donc $y_{(1)} + S_0 \subset S$. ■

4.2.2 Résolution de $y' = Ay$: équation différentielle linéaire à coefficients constants et sans second membre

Rappel : Si $m = 1$, la solution du problème de Cauchy $y' = ay$, $y(t_0) = y_0$ est donnée par $y(t) = y_0 e^{a(t-t_0)}$, $\forall t \in \mathbb{R}$.

Définition 4.2.4. *L'exponentielle de la matrice A est définie par la somme de la série absolument convergente*

$$\exp(A) = \sum_{n=0}^{+\infty} \frac{1}{n!} A^n.$$

Théorème 4.2.5
La solution y du problème de Cauchy $y' = Ay$, $y(t_0) = y_0 \in \mathbb{C}^m$ est donnée par $y(t) = \exp((t-t_0)A) \times y_0$, $\forall t \in \mathbb{R}$.

Démonstration. $y(t_0) = \exp(0 * A) \times y_0 = I * y_0 = y_0$ (OK pour les conditions initiales).

$$\exp(tA) = \sum_{n=0}^{+\infty} \frac{1}{n!} t^n A^n$$

$$\frac{d}{dt}(\exp tA) = \sum_{n=0}^{+\infty} \frac{1}{n!} n t^{n-1} A^n = \sum_{p=0}^{+\infty} \frac{1}{p!} t^p A^{p+1} = Aexp(tA)$$

Par conséquent, $y'(t) = A\exp((t-t_0)A) \times y_0 = Ay(t)$ (OK pour l'équation différentielle).

D'après le théorème de Cauchy Lipschitz, on a trouvé l'unique solution. ■

4.2.3 Résolution de $y' = Ay + b(t)$: équa. diff. linéaire à coefficients constants

Comme l'ensemble de solutions est un espace affine et que l'on sait résoudre $y' = Ay$, il nous suffit de trouver une solution particulière y_p.

La méthode de la variation de la constante. On cherche y_p sous la forme $y_p(t) = \exp(tA) \times v(t)$ où v est supposée dérivable.

$$y_p'(t) = A\exp(tA)v(t) + \exp(tA)v'(t) = Ay_p(t) + \exp(tA)v'(t).$$

Pour que $y_p'(t) = Ay_p(t) + b(t)$, il suffit de choisir v telle que pour tout t, $\exp(tA)v'(t) = b(t)$. On prend donc, pour un certain $t_0 \in \mathbb{R}$, $v(t) = \int_{t_0}^t \exp(-uA)b(u)du$ et on obtient

$$y_p(t) = \int_{t_0}^t \exp((t-u)A)b(u)du.$$

La solution du problème de Cauchy avec conditions initiales (t_0, y_0) est alors

$$y(t) = \exp((t-t_0)A)y_0 + y_p(t) = \exp((t-t_0)A)y_0 + \int_{t_0}^t \exp((t-u)A)b(u)du.$$

Conclusion Pour résoudre les équations différentielles linéaires à coefficients constants, il suffit de savoir calculer :

1. l'exponentielle d'une matrice,
2. l'intégrale d'une fonction de $\mathbb{R}$ dans $\mathbb{C}^m$.

4.3 Méthodes numériques à un pas

4.3.1 Introduction

On veut résoudre le problème de Cauchy

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

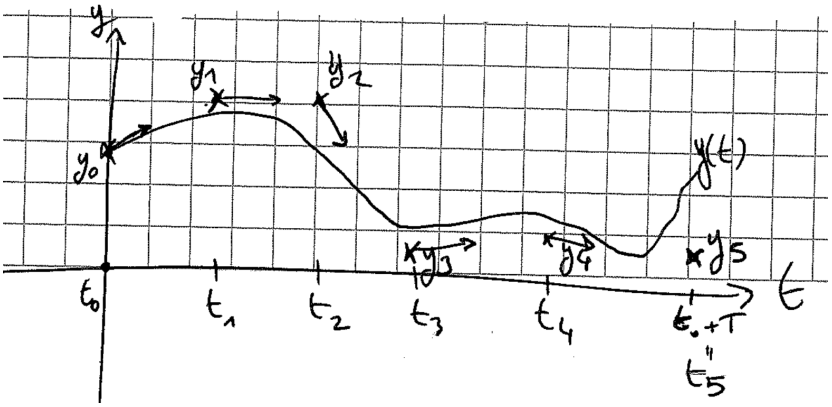
Mais très souvent, on n'arrive pas à calculer la solution exacte. De la même manière que pour les intégrales, on va chercher des solutions approchées.

TABLE 4.1 – **Idée générale des méthodes à 1 pas**

équation différentielle	méthode numérique
$I \subset \mathbb{R}, I = [t_0, t_0 + T]$	discrétisation : $t_0, t_1, \ldots, t_N$ $t_N = t_0 + T$ $h_n = t_{n+1} - t_n$
$y'(t_n)$	$\frac{y_{n+1} - y_n}{h_n}$
$f(t_n, y(t_n))$	$\phi(t_n, y_n, h_n)$
$\begin{cases} y'(t) = f(t, y(t)), \forall t \in I \\ y(t_0) = y_0 \end{cases}$	$\begin{cases} y_{n+1} = y_n + h_n \phi(t_n, y_n, h_n) \\ y_0 \text{ donné} \end{cases}$ y_n est calculable par récurrence

La méthode d'Euler Elle consiste à choisir $\phi(t_n, y_n, h_n) = f(t_n, y_n)$ et donc

$$y_{n+1} = y_n + h_n f(t_n, y_n).$$



4.3.2 Consistance, stabilité, convergence

Définition 4.3.1 (Erreur de consistance locale)**.** *Soit z la solution du problème de Cauchy avec condition initiale $z(t_n) = y_n$*

$$\begin{cases} z'(t) = f(t, z(t)) \\ z(t_n) = y_n \end{cases}$$

L'erreur de consistance locale est définie par $e_n = z(t_{n+1}) - y_{n+1} = z(t_{n+1}) - y_n - \phi(t_n, y_n, h_n)$.

Exemple : La méthode d'Euler, $y_{n+1} = y_n + h_n f(t_n, y_n)$

Si f est C^1,
$z(t_{n+1}) = z(t_n) + (t_{n+1} - t_n)z'(t_n) + \dfrac{(t_{n+1} - t_n)^2}{2} z''(t_n) + o((t_{n+1} - t_n)^2)$.

On a $z(t_n) = y_n$, $z'(t_n) = f(t_n, z(t_n)) = f(t_n, y_n)$ et

$$z''(t_n) = \frac{\partial f}{\partial t}(t_n, z(t_n)) + \underbrace{\frac{\partial f}{\partial z}(t_n, z(t_n))}_{\text{Jacobienne}} \times z'(t_n) \overset{\text{déf}}{=} f^{[1]}(t_n, y_n)$$

Ainsi,

$$\begin{aligned} \|e_n\| &= \|z(t_{n+1}) - y_{n+1}\| = \|y_n + h_n f(t_n, y_n) + \frac{h_n^2}{2} f^{[1]}(t_n, y_n) + o(h_n^2) - y_n - h_n f(t_n, y_n)\| \\ &= \frac{h_n^2}{2} \|f^{[1]}(t_n, y_n)\| + o(h_n^2). \end{aligned}$$

Proposition 4.3.2
L'erreur de consistance locale vérifie $\|e_n\| \leq C h_n^{p+1}$ avec $C > 0$ indépendant de h_n si et seulement si :

- *f est de classe C^p*
- *$\forall l, 0 \leq l \leq p-1, \dfrac{\partial^l \phi}{\partial h^l}(t, y, 0) = \dfrac{1}{l+1} f^{[l]}(t, y)$*
  où $f^{[l]}(t, y) = \Big(\dfrac{d^l}{d\tau^l} (f(\tau, z(\tau))) \Big|_{\substack{z(t) = y \\ z'(\tau) = f(\tau, z(\tau))}} \Big)(t)$

Démonstration.

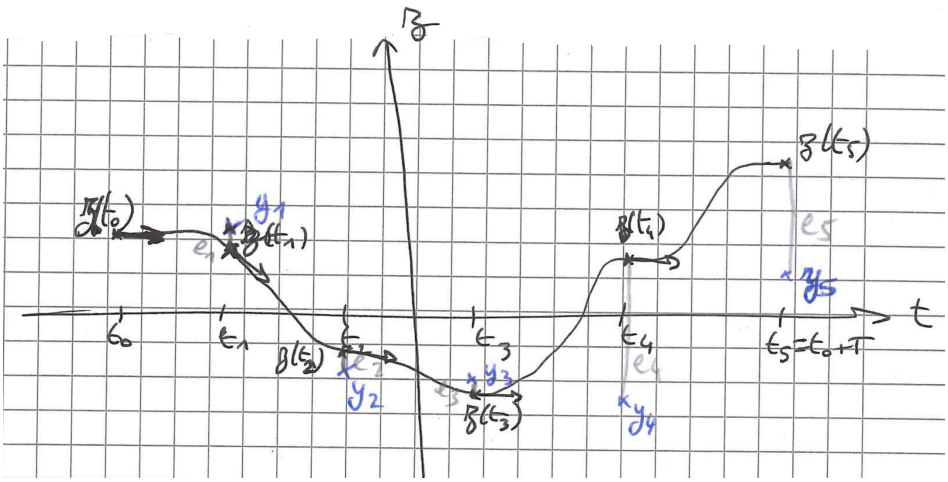
$$\begin{aligned} y_{n+1} &= y_n + h_n \phi(t_n, y_n, h_n) \\ &= y_n + h_n \Big[\phi(t_n, y_n, 0) + h_n \frac{\partial \phi}{\partial h}(t_n, y_n, 0) + \ldots + \frac{h_n^{p-1}}{(p-1)!} \frac{\partial^{p-1} \phi}{\partial h^{p-1}}(t_n, y_n, 0) + \frac{h_n^p}{p!} \frac{\partial^p \phi}{\partial h^p}(t_n, y_n, 0) + o(h_n^p) \Big] \\ z(t_{n+1}) &= z(t_n) + h_n f(t_n, z(t_n)) + \frac{h_n^2}{2} f^{[1]}(t_n, z(t_n)) + \ldots + \frac{h_n^{p+1}}{(p+1)!} f^{[p]}(t_n, z(t_n)) + o(h_n^{p+1}) \\ \|e_n\| &= \|y_{n+1} - z(t_{n+1}\| = \|h_n \sum_{l=0}^{p} \left[\frac{h_n^l}{l!} \frac{\partial^l \phi}{\partial h^l}(t_n, y_n, 0) - \frac{h_n^l}{(l+1)!} f^{[l]}(t_n, z(t_n)) \right] + o(h_n^{p+1})\| \\ &= h_n^{p+1} \| \frac{1}{p!} \frac{\partial^p \phi}{\partial h^p}(t_n, y_n, 0) - \frac{1}{(p+1)!} f^{[p]}(t_n, z(t_n))\| + o(h_n^{p+1}) \end{aligned}$$

■

Définition 4.3.3 (Consistance)**.** *La méthode ϕ est consistante si pour toute solution exacte z, la somme des erreurs de consistance relatives à z tend vers 0 quand $h_{\max} = \max_n h_n$ tend vers 0.*

Soit $e_n = z(t_{n+1}) - z(t_n) - h_n\phi(t_n, z(t_n), h_n)$. La définition signifie que

$$\lim_{h_{\max}\to 0} \sum_{n=0}^{N-1} \|e_n\| = 0.$$



Définition 4.3.4. *La méthode ϕ est stable si il existe une constante $S \geq 0$ indépendante de N, appelée constante de stabilité, telle que pour toutes suites (y_n) et $(\tilde{y}_n)$ définies par*

$$\begin{aligned} y_{n+1} &= y_n + h_n\phi(t_n, y_n, h_n), && 0 \leq n \leq N-1 \\ \tilde{y}_{n+1} &= \tilde{y}_n + h_n\phi(t_n, \tilde{y}_n, h_n) + \epsilon_n, && 0 \leq n \leq N-1 \end{aligned}$$

on ait $\max_{0\leq n\leq N}\|\tilde{y}_n - y_n\| \leq S(\|\tilde{y}_0 - y_0\| + \sum_{n=0}^{N-1}\|\epsilon_n\|)$.

Remarque : Stabilité ⇒ propagation des erreurs maîtrisée

Définition 4.3.5 (Convergence)**.** *La méthode ϕ est convergente si pour toute solution exacte z, la suite (y_n) telle que $y_{n+1} = y_n + h_n\phi(t_n, y_n, h_n)$ vérifie*

$$\max_{0\leq n\leq N}\|y_n - z(t_n)\| \to 0 \quad \textit{quand} \quad y_0 \to z(t_0) \textit{ et } h_{\max} \to 0.$$

La quantité $\max_{0\leq n\leq N}\|y_n - z(t_n)\|$ *s'appelle l'erreur globale*

Théorème 4.3.6
Si la méthode est stable et consistante, alors elle est convergente.

Démonstration. Soit z la solution exacte du problème de Cauchy.

Posons $\tilde{y}_n = z(t_n)$: par définition de l'erreur de consistance locale e_n, on a

$$\tilde{y}_{n+1} = \tilde{y}_n + h_n\phi(t_n, \tilde{y}_n, h_n) + e_n.$$

Comme la méthode est stable de constante de stabilité S, on a

$$\max_{0\leq n\leq N}\|y_n - z(t_n)\| = \max_{0\leq n\leq N}\|y_n - \tilde{y}_n\| \leq S(\|y_0 - z(t_0)\| + \sum_{n=0}^{N-1}\|e_n\|).$$

Comme la méthode est consistante, $\lim_{h_{\max}\to 0}\sum_{n=0}^{N-1}\|e_n\| = 0$. On obtient que

$$\lim_{\substack{h_{\max}\to 0\\ y_0\to z(t_0)}} \max_{0\le n\le N}\|y_n - z(t_n)\| = 0.$$

■

4.3.3 *Condition nécessaire et suffisante de consistance

On sait déjà que si f est C^1 et $\phi(t,y,0) = f(t,y)$, alors $\forall n$, $\|e_n\| \le Ch_n^2$. Pour le résultat suivant, on va supposer uniquement que f est C^0 et on va majorer $\sum_{n=0}^{N}\|e_n\|$.

Théorème 4.3.7
Supposons que $f : \mathbb{R}\times\mathbb{R}^m \to \mathbb{R}^m$ est continue. La méthode à 1 pas définie par ϕ est consistante si et seulement si

$$\forall t \in [t_0, t_0+T], \forall y \in \mathbb{R}^m, \phi(t,y,0) = f(t,y).$$

Démonstration. Soit z une solution exacte du problème de Cauchy et soit e_n l'erreur de consistance locale $e_n = z(t_{n+1}) - z(t_n) - h_n\phi(t_n, z(t_n), h_n)$.

$$\begin{aligned}
z(t_{n+1}) &= z(t_n) + \int_{t_n}^{t_{n+1}} f(t,z(t))dt\\
\|e_n\| &= \|z(t_{n+1}) - z(t_n) - h_n\phi(t_n, z(t_n), h_n)\|\\
&= \|\int_{t_n}^{t_{n+1}} f(t,z(t))dt - \int_{t_n}^{t_{n+1}} \phi(t_n, z(t_n), h_n)dt\|\\
&= \|\int_{t_n}^{t_{n+1}} f(t,z(t))dt - \int_{t_n}^{t_{n+1}} \phi(t,z(t),0)dt + \int_{t_n}^{t_{n+1}} \phi(t,z(t),0) - \phi(t_n,z(t_n),h_n)dt\|\\
&\le \underbrace{\int_{t_n}^{t_{n+1}} \|f(t,z(t)) - \phi(t,z(t),0)\|dt}_{A_n} + \underbrace{\int_{t_n}^{t_{n+1}} \|\phi(t,z(t),0) - \phi(t_n,z(t_n),h_n)\|dt}_{B_n}
\end{aligned}$$

$\sum_{n=0}^{N}\|e_n\| = \sum_{n=0}^{N} A_n + \sum_{n=0}^{N} B_n$
$\sum_{n=0}^{N} A_n$: $\sum_{n=0}^{N} A_n = \sum_{n=0}^{N}\int_{t_n}^{t_{n+1}}\|f(t,z(t)) - \phi(t,z(t),0)\|dt = \int_{t_0}^{T}\|f(t,z(t)) - \phi(t,z(t),0)\|dt$
$\sum_{n=0}^{N} B_n$: ϕ est continue donc $((t,h)\mapsto \phi(t,z(t),h))$ l'est aussi.
Pour tout δ, $[t_0,t_0+T]\times[0,\delta]$ est compact donc $((t,h)\mapsto\phi(t,z(t),h))$ est uniformément continue sur cet ensemble. Cela veut dire que $\forall\epsilon>0$, $\exists\eta>0$ ($\eta\le\delta$) tel que $\forall h_{\max}\le\eta$, on a $\forall n, \forall t\in[t_0,t_0+T], \|\phi(t,z(t),0) - \phi(t_n,z(t_n),h_n)\| \le \epsilon$. Ainsi, $\sum_{n=0}^{N} B_n \le \sum_{n=0}^{N} h_n\epsilon = T\epsilon$.

Au final, $\sum_{n=0}^{N}\|e_n\| = \int_{t_0}^{T}\|f(t,z(t)) - \phi(t,z(t),0)\|dt + T\epsilon$.

Si $\exists J\subset[0;T]$ tel que $f(t,z(t)) \ne \phi(t,z(t),0)$ sur J, alors $\sum_{n=0}^{N}\|e_n\| \not\to 0$.

Si $\forall t$, $f(t,z(t)) = \phi(t,z(t),0)$, alors $\sum_{n=0}^{N}\|e_n\| \to 0$. ■

4.3.4 Condition suffisante de stabilité

Théorème 4.3.8
Si ϕ est lipschitzienne en y de constante Λ, alors la méthode est stable avec constante de stabilité $e^{\Lambda T}$.

Dit autrement, si $\forall t \in [t_0, t_0+T]$, $\forall y_1, y_2 \in \mathbb{R}^m$, $\forall h \geq 0$, on a $\|\phi(t, y_1, h) - \phi(t, y_2, h)\| \leq \Lambda\|y_1 - y_2\|$, alors pour $y_{n+1} = y_n + h_n\phi(t_n, y_n, h_n)$ et $\tilde{y}_{n+1} = \tilde{y}_n + h_n\phi(t_n, \tilde{y}_n, h_n) + \epsilon_n$, on a

$$\max_{0\leq n\leq N}\|\tilde{y}_n - y_n\| \leq e^{\Lambda T}\Big(\|\tilde{y}_0 - y_0\| + \sum_{n=0}^{N}\|e_n\|\Big).$$

Lemme 4.3.9 (Gronwall, cas discret)
Soient $(h_n), (\theta_n), (\epsilon_n)$ des suites de $\mathbb{R}_+$ telles que pour tout n,

$$\theta_{n+1} \leq (1 + \Lambda h_n)\theta_n + \epsilon_n.$$

Alors pour tout n,

$$\theta_n \leq e^{\Lambda(t_n - t_0)}\theta_0 + \sum_{i=1}^{n-1} e^{\Lambda(t_n - t_i)}\epsilon_i.$$

Démonstration du lemme. Preuve par récurrence.

Pour $n = 0$, il suffit de vérifier que $\theta_0 \leq \theta_0$.

Supposons que $\theta_n \leq e^{\Lambda(t_n-t_0)}\theta_0 + \sum_{i=1}^{n-1} e^{\Lambda(t_n-t_i)}\epsilon_i$.

$$\theta_{n+1} \leq (1 + \Lambda h_n)\theta_n + \epsilon_n \leq (1 + \Lambda h_n)\Big(e^{\Lambda(t_n-t_0)}\theta_0 + \sum_{i=1}^{n-1} e^{\Lambda(t_n-t_i)}\epsilon_i\Big) + \epsilon_n$$

Mais comme $1 + \Lambda h_n \leq e^{\Lambda h_n} = e^{\Lambda(t_{n+1}-t_n)}$, et $\epsilon_n \leq e^{\Lambda(t_{n+1}-t_n)}\epsilon_n$,

$$\theta_{n+1} \leq e^{\Lambda(t_{n+1}-t_n+t_n-t_0)}\theta_0 + \sum_{i=1}^{n-1} e^{\Lambda(t_{n+1}-t_n+t_n-t_i)}\epsilon_i + e^{\Lambda(t_{n+1}-t_n)}\epsilon_n = e^{\Lambda(t_{n+1}-t_0)}\theta_0 + \sum_{i=1}^{n} e^{\Lambda(t_{n+1}-t_i)}\epsilon_i.$$

■

Démonstration du théorème. Considérons deux suites (y_n) et $(\tilde{y}_n)$ telles que

$$y_{n+1} = y_n + h_n\phi(t_n, y_n, h_n)$$
$$\tilde{y}_{n+1} = \tilde{y}_n + h_n\phi(t_n, \tilde{y}_n, h_n) + \epsilon_n$$

Par différence, on obtient $\|y_{n+1} - \tilde{y}_{n+1}\| \leq \|y_n - \tilde{y}_n\| + h_n\Lambda\|y_n - \tilde{y}_n\| + \|\epsilon_n\|$.

On peut utiliser le lemme de Gronwall avec $\theta_n = \|y_n - \tilde{y}_n\|$, $h_n = h_n$ et $\epsilon_n = \|\epsilon_n\|$.
Cela donne $\forall n$, $\|y_n - \tilde{y}_n\| \leq e^{\Lambda(t_n-t_0)}\|y_0 - \tilde{y}_0\| + \sum_{i=0}^{n-1} e^{\Lambda(t_n-t_i)}\|\epsilon_i\|$
et donc $\forall n, \|y_n - \tilde{y}_n\| \leq e^{\Lambda T}\Big(\|y_0 - \tilde{y}_0\| + \sum_{i=0}^{n}\|\epsilon_n\|\Big)$, vu que $e^{\Lambda(t_n-t_i)} \leq e^{\Lambda T}$. ■

4.3.5 Majoration de l'erreur globale

- Si la méthode est d'ordre p et f est de classe C^p, $\|e_n\| \leq Ch_n^{p+1}$ où C est une constante indépendante de n et de $h_n \leq \delta$.

- Si la méthode est stable de constante de stabilité S,

$$\max_{0\leq n\leq N}\|y_n - z(t_n)\| \leq S\Big(\|y_0 - z(t_0)\| + \sum_{n=0}^{N}\|e_n\|\Big).$$

En combinant les deux, on obtient $\sum_{n=0}^{N}\|e_n\| \leq \sum_{n=0}^{N} Ch_n^{p+1} \leq Ch_{\max}^p \sum_{n=0}^{N} h_n = CTh_{\max}^p$ et

$$\max_{0\leq n\leq N}\|y_n - z(t_n)\| \leq S\|y_0 - z(t_0)\| + SCTh_{\max}^p.$$

Si ϕ est lipschitzienne en y de constante Λ,

$$\max_{0\leq n\leq N}\|y_n - z(t_n)\| \leq e^{\Lambda T}\|y_0 - z(t_0)\| + CTe^{\Lambda T}h_{\max}^p.$$

4.4 Méthodes de Runge-Kutta

4.4.1 Principe

Le problème de Cauchy à résoudre est

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

$[t_0, t_0 + T]$ est discrétisé en t_0, t_1, …, t_N.

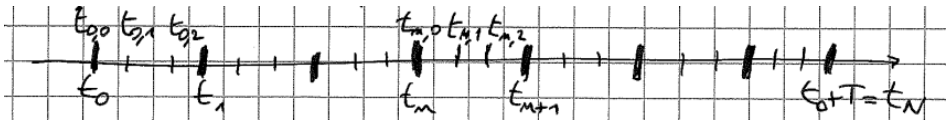
Soit z la solution exacte du problème de Cauchy : $z(t_{n+1}) = z(t_n) + \int_{t_n}^{t_{n+1}} f(t, z(t))dt$

La méthode d'Euler $y_{n+1} = y_n + h_n f(t_n, y_n)$ correspond à approcher $\int_{t_n}^{t_{n+1}} f(t, z(t))dt$ par les rectangles à gauche, de manière successive.

Les méthodes de Runge-Kutta correspondent à schéma d'intégration quelconque.

Il faut des points intermédiaires entre t_n et t_{n+1} :

$$t_{n,i} = t_n + c_i h_n, \qquad 1 \leq i \leq q \qquad c_i \in [0, 1]$$



Rappel : Soit $g : \mathbb{R} \to \mathbb{R}^m$. $\int_0^1 g(t)dt \approx \sum_{j=1}^{q} b_j g(c_j)$ où $\sum_{j=1}^{q} b_j = 1$.

Les b_j sont les coefficients donnés par la technique d'approximation de l'intégrale.

Difficulté : On aimerait prendre $g(t) = f(t, z(t))$ mais z n'est pas connu. Ainsi :

- On approche $z(t_n)$ par y_n.
- On approche $z(t_{n,i}) = z(t_n) + \int_{t_n}^{t_{n,i}} f(t, z(t))dt = z(t_n) + \int_0^{c_i} f(t_n + uh_n, z(t_n + uh_n))du$ par $y_{n,i} = y_n + h_n \sum_{j=1}^{i-1} a_{i,j} f(t_{n,j}, y_{n,j}), \quad 1 \leq i \leq q$ où $\sum_{j=1}^{i-1} a_{i,j} = c_i$.
- On approche $z(t_{n+1})$ par $y_{n+1} = y_n + h_n \sum_{j=1}^{q} b_j f(t_{n,j}, b_{n,j})$ où $\sum_{j=1}^{q} b_j = 1$.

Notez que les techniques d'intégration peuvent être toutes différentes.

On obtient l'algorithme suivant :

$$\begin{cases} \forall i \in \{1, \dots q\} \begin{cases} t_{n,i} = t_n + c_i h_n \\ y_{n,i} = y_n + h_n \sum_{j=1}^{i-1} a_{i,j} p_{n,j} \qquad t_{n+1} = t_n + h_n \\ p_{n,i} = f(t_{n,i}, y_{n,i}) \end{cases} \\ y_{n+1} = y_n + h_n \sum_{j=1}^{q} b_j p_{n,j} \end{cases}$$

L'algorithme est bien défini de manière explicite car la somme qui définit $y_{n,i}$ n'utilise que les $y_{n,j}$ pour $j < i$. On peut le représenter par le tableau

$$\begin{array}{c|ccccc} c_1 & 0 & 0 & \dots & 0 & 0 \\ c_2 & a_{2,1} & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \ddots & \vdots \\ c_q & a_{q,1} & a_{q,2} & \dots & a_{q,q-1} & 0 \\ \hline & b_1 & b_2 & \dots & b_{q-1} & b_q \end{array}$$

Chaque ligne correspond à une méthode d'intégration approchée.

Il y a des contraintes sur les coefficients : $c_i = \sum_{j=1}^{i-1} a_{i,j}$, $1 = \sum_{j=1}^{q} b_j$, $c_1 = 0$, $t_{n,1} = t_n$, $y_{n,1} = y_n$, $p_{n,1} = f(t_n, y_n)$.

4.4.2 Exemples

$\underline{q = 1}$

Le seul choix possible est $\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$.

$c_1 = 0$, $a_{1,1} = 0$, $b_1 = 1$.

L'algorithme est donné par $p_{n,1} = f(t_n, y_n)$, $t_{n+1} = t_n + h_n$, $y_{n+1} = y_n + h_n p_n$. Il s'agit de la méthode d'Euler.

$\underline{q = 2}$

Pour $\alpha \in]0; 1]$, et $\beta \in \mathbb{R}$, on a la méthode définie par le tableau : $\begin{array}{c|cc} 0 & 0 & 0 \\ \alpha & \alpha & 0 \\ \hline & 1-\beta & \beta \end{array}$

Exercice : Pour quelles valeurs de α et β la méthode est-elle d'ordre 2 ?

- Pour $\alpha = \frac{1}{2}$ et $\beta = 1$, on a la méthode du point milieu :
  $p_{n,1} = f(t_n, y_n)$
  $t_{n,2} = t_n + \frac{1}{2}h_n$
  $y_{n,2} = y_n + \frac{1}{2}h_n p_{n,1}$
  $p_{n,2} = f(t_{n,2}, y_{n,2})$
  $t_{n+1} = t_n + h_n$
  $y_{n+1} = y_n + h_n p_{n,2}$
  ie : $y_{n+1} = y_n + h_n f(t_n + \frac{1}{2}h_n, y_n + \frac{1}{2}h_n f(t_n, y_n))\Big)$
- Pour $\alpha = 1$ et $\beta = \frac{1}{2}$, on a la méthode de Heun :
  $p_{n,1} = f(t_n, y_n)$
  $t_{n,2} = t_n + h_n$
  $y_{n,2} = y_n + h_n p_{n,1}$
  $p_{n,2} = f(t_{n,2}, y_{n,2})$
  $t_{n+1} = t_n + h_n$
  $y_{n+1} = y_n + \frac{1}{2}h_n p_{n,1} + \frac{1}{2}h_n p_{n,1}$

  La méthode de Heun peut aussi s'écrire
  $y_{n+1} = y_n + h_n\Big(\frac{1}{2}f(t_n, y_n) + \frac{1}{2}f(t_{n+1}, y_n + h_n f(t_n, y_n))\Big)$.
  Cependant cette formule fait intervenir 3 évaluations de f alors que le fait de stocker $f(t_n, y_n)$ dans $p_{n,1}$ permet de n'en faire que 2.

RK4 (Runge-Kutta classique avec $q = 4$)
L'algorithme s'écrit
$p_{n,1} = f(t_n, y_n)$
$t_{n,2} = t_n + \frac{1}{2}h_n$
$y_{n,2} = y_n + \frac{1}{2}h_n p_{n,1}$
$p_{n,2} = f(t_{n,2}, y_{n,2})$

$t_{n,3} = t_{n,2}$
$y_{n,3} = y_n + \frac{1}{2}h_n p_{n,2}$
$p_{n,3} = f(t_{n,3}, y_{n,3})$

0	0	0	0	0	
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	$\rightarrow$ rectangles à gauche
$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	$\rightarrow$ rectangles à droite
1	0	0	1	0	$\rightarrow$ point milieu
	$\frac{1}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{1}{6}$	$\rightarrow$ Simpson

$t_{n,4} = t_n + h_n$
$y_{n,4} = y_n + h_n p_{n,3}$
$p_{n,4} = f(t_{n,4}, y_{n,4})$

$t_{n+1} = t_n + h_n$
$y_{n+1} = y_n + h_n(\frac{1}{6}p_{n,1} + \frac{2}{6}p_{n,2} + \frac{2}{6}p_{n,3} + \frac{1}{6}p_{n,4})$

4.4.3 Stabilité des méthodes de Runge-Kutta

Ce sont des méthodes à un pas : $y_{n+1} = y_n + h_n\phi(t_n, y_n, h_n)$
où $\phi(t, y, h) = \sum_{j=1}^{a} b_j f(t + c_j h, y_j)$ et $y_i = y + h\sum_{j=1}^{i-1} a_{i,j} f(t + c_j h, y_j)$

Pour l'étude de la stabilité, on définit pour tout i la fonction $\bar{y}_i$ telle que $\bar{y}_i(t, y, h) = y + h\sum_{j=1}^{i-1} a_{i,j} f(t + c_j h, \bar{y}_j(t, y, h))$.

Lemme 4.4.1
Si f est k-lipschitzienne en y alors avec $\alpha = \max_{1\leq i\leq q}\left(\sum_{j=1}^{i} |a_{i,j}|\right)$, on a $\forall y, z \in \mathbb{R}^m$,-
$\forall i \in \{1, \ldots, q\}, \|\bar{y}_i(t, y, h) - \bar{y}_i(t, z, h)\| \leq (1 + (\alpha kh) + (\alpha kh)^2 + \ldots (\alpha kh)^{i-1})\|y - z\|$

Démonstration. On fait une récurrence sur i. Pour $i = 1$, on a $\bar{y}_1(t, y, h) = y$ et $\bar{y}_1(t, z, h) = z$ donc c'est bon.

Supposons l'inégalité vraie pour tout $j < i$.

$$\begin{aligned}
\|\bar{y}_i(t, y, h) - \bar{y}_i(t, z, h)\| &\leq \left\| y + h\sum_{j=1}^{i-1} a_{i,j} f(t + c_j h, \bar{y}_j(t, y, h)) - z - h\sum_{j=1}^{i-1} a_{i,j} f(t + c_j h, \bar{y}_j(t, z, h)) \right\| \\
&\leq \|y - z\| + h\sum_{j=1}^{i-1} |a_{i,j}| \, \|f(t + c_j h, \bar{y}_j(t, y, h)) - f(t + c_j h, \bar{y}_j(t, z, h))\| \\
&\leq \|y - z\| + h\sum_{j=1}^{i-1} |a_{i,j}| k \|\bar{y}_j(t, y, h) - \bar{y}_j(t, z, h)\|
\end{aligned}$$

On utilise l'hypothèse de récurrence pour obtenir

$$\begin{aligned}\|\bar{y}_i(t,y,h)-\bar{y}_i(t,z,h)\| &\leq \|y-z\| + h\sum_{j=1}^{i-1}|a_{i,j}|k(1+(\alpha kh)+(\alpha kh)^2+\ldots(\alpha kh)^{j-1})\|y-z\| \\ &\leq \|y-z\| + h\sum_{j=1}^{i-1}|a_{i,j}|k(1+(\alpha kh)+(\alpha kh)^2+\ldots(\alpha kh)^{i-2})\|y-z\| \\ &\leq \big(1+(\alpha kh)+(\alpha kh)^2+\ldots(\alpha kh)^{i-1}\big)\|y-z\| \qquad \blacksquare\end{aligned}$$

Théorème 4.4.2
Les méthodes de Runge Kutta sont stables, avec constante de stabilité $S = e^{\Lambda T}$ *où* $\Lambda = k\sum_{j=1}^{q}|b_j|(1+(\alpha kh_{\max})+\ldots+(\alpha kh_{\max})^{j-1})$.

Démonstration. Il suffit de montrer que ϕ est Λ-lipschitzienne en y.

Soient $y, z \in \mathbb{R}^m$, $t \in \mathbb{R}$ et $h > 0$.

$$\begin{aligned}\|\phi(t,y,h)-\phi(t,z,h)\| &= \|\sum_{j=1}^{q} b_j\big(f(t+c_jh,\bar{y}_j(t,y,h)-f(t+c_jh,\bar{y}_j(t,z,h))\big)\| \\ &\leq \sum_{j=1}^{q}|b_j|k\|\bar{y}_j(t,y,h)-\bar{y}_j(t,z,h)\| \\ &\leq k\sum_{j=1}^{q}|b_j|(1+(\alpha kh_{\max})+\ldots+(\alpha kh_{\max})^{j-1}) \qquad \blacksquare\end{aligned}$$

Remarquons que si $h_{\max}$ est petit devant $\frac{1}{\alpha k}$, alors S est de l'ordre de e^{kT}, ce qui est comparable à la méthode d'Euler.

4.4.4 Ordre des méthodes de Runge-Kutta

On rappelle que

- $\phi(t,y,h) = \sum_{j=1}^{q} b_j f(t+c_jh,\bar{y}_j(t,y,h))$
  où $\bar{y}_j(t,y,h) = y + h\sum_{j=1}^{i-1} a_{i,j}f(t+c_jh,\bar{y}_j(t,y,h))$.
- L'ordre de la méthode est p au moins si et seulement si $\forall l \leq p-1, \forall t \in \mathbb{R}, \forall y \in \mathbb{R}^m$,

$$\frac{\partial^l \phi}{\partial h^l}(t,y,0) = \frac{1}{l+1}f^{[l]}(t,y)$$

1. $\frac{\partial^0\phi}{\partial h^0}(t,y,0) = \phi(t,y,0) = f(t,y)$ car $\bar{y}_i(t,y,0) = y$ et $\sum_{j=1}^{q} b_j = 1$. Ainsi les méthode de Runge-Kutta sont toujours d'ordre 1 et donc consistantes.
2. $\frac{\partial^1\phi}{\partial h^1}(t,y,h) = \sum_{j=1}^{q} b_j\Big(\frac{\partial f}{\partial t}(t+c_jh,\bar{y}_j(t,y,h))\times c_j + \frac{\partial f}{\partial y}(t+c_jh,\bar{y}_j(t,y,h))\times\frac{\partial\bar{y}_j}{\partial h}(t,y,h)\Big)$
   $\frac{\partial\bar{y}_i}{\partial h}(t,y,h) = \sum_{j<i} a_{i,j}f(t+c_jh,\bar{y}_j(t,y,h)) + h\sum_{j<i}a_{i,j}\Big(\frac{\partial f}{\partial t}(-)c_j + \frac{\partial f}{\partial y}(-)\frac{\partial\bar{y}_j}{\partial h}(-)\Big)$
   Ainsi $\frac{\partial\bar{y}_i}{\partial h}(t,y,0) = \sum_{j<i}a_{i,j}f(t,y) = c_if(t,y)$.
   $\frac{\partial^1\phi}{\partial h^1}(t,y,0) = \sum_{j=1}^{q} b_j\frac{\partial f}{\partial t}(t,y)\times c_j + \frac{\partial f}{\partial y}(t,y)\times f(t,y)\times c_j = \big(\sum_{j=1}^{q} b_jc_j\big)f^{[1]}(t,y)$
   On en déduit que la méthode est d'ordre 2 si et seulement si $\sum_{j=1}^{q} b_jc_j = \frac{1}{2}$.

3. Il faut dériver encore une fois.

$$\frac{\partial^2\phi}{\partial h^2}(t,y,h) = \sum_{j=1}^{q} b_j \Bigg(\frac{\partial f}{\partial t^2}(t+c_jh, \bar{y}_j(t,y,h))c_j^2 + 2c_j \frac{\partial^2 f}{\partial t \partial y}(t+c_jh, \bar{y}_j(t,y,h)) \frac{\partial \bar{y}_j}{\partial y}(t,y,h)$$
$$+ \frac{\partial f}{\partial y}(t+c_jh, \bar{y}_j(t,y,h)) \frac{\partial^2 \bar{y}_j}{\partial h^2}(t,y,h) + \Big\langle \frac{\partial^2 f}{\partial y^2}(t+c_jh, \bar{y}_j(t,y,h)) \frac{\partial \bar{y}_j}{\partial h}(t,y,h), \frac{\partial \bar{y}_j}{\partial h}(t,y,h) \Big\rangle \Bigg)$$

$$\frac{\partial^2 \bar{y}_i}{\partial h^2}(t,y,h) = 2\sum_{j<i} a_{i,j} \Big(\frac{\partial f}{\partial t}(-)c_j + \frac{\partial f}{\partial y}(-) \frac{\partial \bar{y}_j}{\partial h}(-) \Big) + h \sum_{j<i} a_{i,j} \Big(\frac{\partial^2 f}{\partial y^2} \dots \Big)$$
$$\frac{\partial^2 \bar{y}_i}{\partial h^2}(t,y,0) = 2\sum_{j<i} a_{i,j} c_j f^{[1]}(t,y)$$

$$\frac{\partial^2\phi}{\partial h^2}(t,y,0) = \sum_{j=1}^{q} b_j \Bigg(\frac{\partial f}{\partial t^2}(t,y)c_j^2 + 2c_j^2 \frac{\partial^2 f}{\partial t \partial y}(t,y) f(t,y)$$
$$+ \frac{\partial f}{\partial y}(t,y) 2\sum_{l<j} a_{j,l} f^{[1]}(t,y) + \Big\langle \frac{\partial^2 f}{\partial y^2}(t,y)c_j f(t,y), c_j f(t,y) \Big\rangle \Bigg)$$
$$= \sum_{j=1}^{q} b_j c_j^2 \Big(\frac{\partial f}{\partial t^2}(t,y) + 2\frac{\partial^2 f}{\partial t \partial y}(t,y) f(t,y) + \Big\langle \frac{\partial^2 f}{\partial y^2}(t,y) f(t,y), f(t,y) \Big\rangle \Big)$$
$$+ 2\sum_{j=1}^{q}\sum_{l=1}^{q} b_j a_{j,l} c_l \frac{\partial f}{\partial y}(t,y) f^{[1]}(t,y)$$

D'un autre côté,

$$f^{[2]}(t,y) = \frac{\partial f}{\partial t^2}(t,y) + 2\frac{\partial^2 f}{\partial t \partial y}(t,y) f(t,y) + \Big\langle \frac{\partial^2 f}{\partial y^2}(t,y) f(t,y), f(t,y) \Big\rangle + \frac{\partial f}{\partial y}(t,y) f^{[1]}(t,y)$$

Pour avoir $\frac{\partial^2\phi}{\partial h^2}(t,y,0) = \frac{1}{3} f^{[2]}(t,y)$, il faut que $\sum_{j=1}^{q} b_j c_j^2 = \frac{1}{3}$ et $\sum_{i,j=1}^{q} b_i a_{i,j} c_j = \frac{1}{6}$.
Ainsi la méthode est d'ordre 3 $\Leftrightarrow \sum_{j=1}^{q} b_j c_j = \frac{1}{2}$, $\sum_{j=1}^{q} b_j c_j^2 = \frac{1}{3}$ et $\sum_{i,j=1}^{q} b_i a_{i,j} c_j = \frac{1}{6}$.

4. On peut obtenir des résultats similaires pour les ordres suivants. Pour l'ordre 4 :

$$\text{ordre 4} \Leftrightarrow \sum_{j=1}^{q} b_j c_j = \frac{1}{2}; \sum_{j=1}^{q} b_j c_j^2 = \frac{1}{3}; \sum_{i,j=1}^{q} b_i a_{i,j} c_j = \frac{1}{6}; \sum_{j=1}^{q} b_j c_j^3 = \frac{1}{4};$$
$$\sum_{i,j=1}^{q} b_i a_{i,j} c_j^2 = \frac{1}{12}; \sum_{i,j=1}^{q} b_i c_i a_{i,j} c_j = \frac{1}{8} \text{ et } \sum_{i,j,k=1}^{q} b_i a_{i,j} a_{j,k} c_k = \frac{1}{12}$$

Exemple : La méthode de Runge-Kutta classique est d'ordre 4.

0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	0	0
1	0	0	1	0
	$\frac{1}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{1}{6}$

1. On a bien $\sum_j b_j = 1$
2. $\sum_j b_j c_j = \frac{1}{6} \times 0 + \frac{2}{6} \times \frac{1}{2} + \frac{2}{6} \times \frac{1}{2} + \frac{1}{6} \times 1 = \frac{1}{2}$
3. $\sum_j b_j c_j^2 = \frac{1}{6} \times 0^2 + \frac{2}{6} \times \frac{1}{4} + \frac{2}{6} \times \frac{1}{4} + \frac{1}{6} \times 1 = \frac{1}{3}$
   $\sum_{i,j} b_i a_{i,j} c_j = b_2 a_{2,1} c_1 + b_3 a_{3,2} c_2 + b_4 a_{4,3} c_3 = \frac{2}{6} \times \frac{1}{2} \times 0 + \frac{2}{6} \times \frac{1}{2} \times \frac{1}{2} + \frac{1}{6} \times 1 \times \frac{1}{2} = \frac{1}{6}$
4. $\sum_j b_j c_j^3 = \ldots$ vérification par ordinateur.

Bibliographie

Jean-Pierre Demailly. *Analyse numérique et équations différentielles*. EDP sciences, 2012.

Alfio Maria Quarteroni, Riccardo Sacco, and Fausto Saleri. *Méthodes numériques : algorithmes, analyse et applications*. Springer Science & Business Media, 2008.