Algorithmique Numérique

Alain Yger

Institut de Mathématiques, Université Bordeaux 1, Talence 33405, France
E-mail address: Alain.Yger@math.u-bordeaux1.fr

Version du 11 décembre 2012.

Résumé. Ce cours correspond à l'enseignement dispensé en 2010-2011 dans l'UE MHT632 ≪ Algorithmique Numérique ≫ du parcours Math-Info. Ce cours approfondit au niveau L3 le cours de L2 d'≪ Initiation au Calcul Scientifique et Symbolique ≫ (MHT304, [**Y1**]). Plusieurs chapitres de l'ouvrage [**MathAp**] (en particulier les chapitres 1,2,3,9,10) ont servi de référence pour la rédaction du cours et peuvent être utilisés pour des approfondissements.

Table des matières

CHAPITRE 1

Représentation des nombres en machine

La rédaction de ce chapitre s'appuie sur les éléments déjà présentés dans le chapitre 1 du cours de MHT304 [**Y1**] ainsi que sur la présentation enrichie de nombreux exemples faite par P. Zimmermann dans [**Zim**].

1.1. Le codage en virgule flottante

Étant donné un entier strictement positif β (dit « base »), tout nombre entier naturel se décompose « en base β » de manière unique sous la forme

$$n = \sum_{j=0}^{N} b_j \beta^j, \quad b_j \in \{0, ..., \beta - 1\}.$$

Si le développement en base β le plus familier est celui qui correspond à $\beta = 10$ (développement *décimal*), le plus en phase avec le calcul machine est le développement en base $\beta = 2$ (développement « binaire ») qui n'utilise que deux symboles :

- 0= l'interrupteur est fermé, *i.e* le courant ne passe pas ;
- 1= l'interrupteur est ouvert, *i.e* le courant passe.

Par exemple, en base 2,

$$19 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2 + 1 = [1\ 0\ 0\ 1\ 1].$$

L'encodage d'un réel en machine, dit *en virgule flottante*, se fait suivant le standard `IEEE754` (1985, révisé en 2008), soit dans l'un des trois formats binaires `binary32`, `binary64`, `binary128`, soit dans l'un des deux formats décimaux que sont `decimal64`, `decimal128`.

Dans un des systèmes binaires (on prendra comme exemple `binary64`, dit *système en double précision* binaire), un nombre réel x (ou plutôt une valeur approchée $\overline{x}$ de ce nombre réel) est encodé sur $1 + 11 + 52 = 64$ bits [1] :

- le premier bit est réservé pour le *signe* du nombre [2] ; on note $s \in \{0, 1\}$ la valeur de ce bit ;
- les 11 bits suivants servent à encoder l'*exposant*, *i.e* l'entier $e \in \mathbb{Z}$ défini (lorsque x est non nul) par le fait que $2^{-e}|x| \in [1/2, 1[$, ce qui signifie que le nombre $2^{-e}|x|$ admet un unique développement binaire propre (c'est-à-dire dont les coefficients ne sont pas tous égaux à 1)

$$2^{-e}|x| = \frac{b_0}{2} + \frac{b_1}{2^2} + \cdots + \frac{b_k}{2^{k+1}} + \cdots, \ b_0 = 1, \ b_j \in \{0, 1\} \ \forall\, j \in \mathbb{N}^* \ ; \tag{1.1}$$

1. En *simple précision*, soit dans `binary32`, le découpage est $1 + 8 + 23 = 32$ bits, tandis qu'en *quadruple précision*, soit dans `binary128`, le découpage est $1 + 15 + 112 = 128$ bits.

2. On note à ce propos qu'une ambiguité existe pour $x = 0$ et que la machine nous force à distinguer -0 et $+0$.

la valeur codée avec ces 11 bits est donc un entier $0 \leq v \leq 2^{11} - 1 = 2047$ et l'exposant $e = v - 1023$ peut prendre toute valeur entière entre -1023 ($v = 0$) et 1024 ($v = 2047$) ;
- les 52 bits restants sont utilisés pour coder le mot $[b_1 \cdots b_{52}]$, mot de 52 lettres, chacune valant 0 ou 1 ; le mot $[1\, b_1 \cdots b_{52}]$ est appelé *mantisse*[3] de x.

Les nombres -0 et $+0$ sont donc encodés respectivement avec $v = 0$ (tous les bits $b_1, ..., b_{52}$ étant mis à 0) tandis que $\pm\infty$ et `NaN` (« Not A Number ») sont encodés avec $v = 2047$ (respectivement lorsque tous les bits $b_1, ..., b_{52}$ sont mis à zéro ou non).

EXEMPLE 1.1. Par exemple, pour π :

$$\pi \simeq \frac{7074237752028440}{2^{51}} = \frac{2^{52} + 2570638124657944}{2^{51}} = 2\Big(1 + \frac{N}{2^{52}}\Big)$$

le numérateur $N = 2570638124657944 \leq 2^{51}$ s'exprimant ici comme un mot de 52 caractères (0 ou 1) en base 2. On a donc $s = 0$, $e = 1$, c'est-à-dire $v = 1024$, le mot $[b_1 \cdots b_{52}]$ correspondant à l'écriture en base 2 du nombre

$$N = 2570638124657944 \leq 2^{51}.$$

L'encodage de π sera ainsi (si l'on respecte l'ordre des bits qui a été indiqué) l'écriture en base 2 du nombre

$$\begin{aligned} & s \times 2^{63} + v \times 2^{52} + N = 4614256656552045848 \\ = \quad & [0100000000001001001000011111101101010100010001000010110100011000] \end{aligned}$$

Si N_b est le nombre de bits impliqués dans le codage de la mantisse ($N_b = 52$ en double précision, $N_b = 23$ en simple précision, $N_b = 112$ en quadruple précision), l'erreur d'arrondi entre une vraie mantisse b et la mantisse « codée » $\overline{b}$ est donc majorée en module par

$$|b - \overline{b}| \leq 2^{-N_b - 1}$$

(on « arrondit »au nombre codable avec $N_b + 1$ chiffres le plus proche, qu'il soit plus petit ou plus grand que m, exactement comme on le fait en décimal). L'erreur relative commise lorsque l'on arrondit $x = \pm b \times 2^e$ en $\overline{x} = \pm\overline{b} \times 2^e = \pm[1\, b_1 \cdots b_{N_b}] \times 2^e$ est donc majorée par

$$\frac{|x - \overline{x}|}{|x|} = \frac{|b - \overline{b}|}{b} \leq \frac{|b - \overline{b}|}{\frac{1}{2}} \leq 2 \times 2^{-N_b - 1} = 2^{-N_b}\,.$$

Cette erreur relative 2^{-N_b} (où N_b est le nombre de bits utilisés pour coder la mantisse), est appelée *erreur machine* ; c'est elle qui sera responsable des *erreurs d'arrondi* dans les calculs (voir la section suivante).

1.2. Arrondis dans les calculs entre flottants

On fixe ici une précision (simple, double ou quadruple), donc un entier N_b (23, 52 ou 115 suivant que l'on est en simple, double ou quadruple) et un entier M_v ($2^8 - 1$, $2^{11} - 1$, $2^{15} - 1$ suivant que l'on est en simple, double ou quadruple

3. Le premier bit matérialisé en $b_0 = 1$ dans (1.1) est dit *bit caché* ; mettre ce bit à 0 conduit à la représentation des nombres *dénormalisés* ou *sous-normaux* (*subnormal*), bien utiles dans les opérations pour éviter par exemple une division par zéro conduisant brutalement à $\pm\infty$.

précision). Si y est un nombre flottant, *i.e* un réel du type $y = (-1)^s[1\, b_1\, \ldots\, b_{N_b}]\, 2^e$ avec $s \in \{0,1\}$ et e tel que $0 \leq v \leq M_v$, on pose

$$\mathrm{ulp}(y) := 2^{e-N_b}.$$

Cette notation (anglosaxonne) vaut pour « *Unit in the Last Place* ».

DÉFINITION 1.2 (les cinq modes d'arrondi correct du standard `IEEE 754`). On dit que y est *arrondi au plus proche* du nombre réel x

- avec arrondi pair (`roundTiesToEven`) si y est un nombre flottant tel que
$$|y - x| \leq \frac{\mathrm{ulp}(y)}{2}$$
avec la convention que si x est exactement la valeur médiane entre deux flottants y_1 et y_2, alors on privilégie celui dont la mantisse est paire ;
- avec arrondi *away* (`roundTiesToAway`) si y est un nombre flottant tel que
$$|y - x| \leq \frac{\mathrm{ulp}(y)}{2}$$
avec la convention que si x est exactement la valeur médiane entre deux flottants y_1 et y_2, alors on privilégie y_2 si $x > 0$, y_1 si $x < 0$.

On dit que y est un *arrondi dirigé* de x

- vers 0 (`roundTowardZero`) si $|y - x| \leq \mathrm{ulp}(y)$ et $|y| \leq |x|$;
- vers $-\infty$ (`roundTowardNegative`) si $|y - x| \leq \mathrm{ulp}(y)$ et $y \leq x$;
- vers $+\infty$ (`roundTowardPositive`) si $|y - x| \leq \mathrm{ulp}(y)$ et $y \geq x$.

Les opérations arithmétiques classiques que l'on introduit entre nombres flottants sont l'addition, la soustraction, la multiplication et la division, que l'on peut implémenter en flottant comme :

$$\begin{aligned}
(a,b) \mapsto a \oplus b &:= \text{arrondi au plus près de } a+b \\
(a,b) \mapsto a \ominus b &:= \text{arrondi au plus près de } a-b \\
(a,b) \mapsto a \otimes b &:= \text{arrondi au plus près de } a \times b \\
(a,b) \mapsto a \oslash b &:= \text{arrondi au plus près de } a/b.
\end{aligned}$$

À la place du choix qui est fait ici, on peut choisir l'un des cinq modes d'arrondi proposés dans la Définition 1.2.

Aux quatre opérations algébriques mentionnées, il faut ajouter la prise de racine carrée et les conversions binaire/décimal.

Le standard `IEEE 754` impose que toutes ces opérations algébriques soient conduites (comme indiqué) avec l'un des cinq modes d'arrondi proposés dans la Définition 1.2. On dit alors que les six opérations mentionnées ont des implémentations *correctement arrondies*. Cette exigence est dite aussi d'*arrondi correct*. Elle est également recommandée dans l'usage des fonctions transcendantes log, exp, sin, cos, tan, atan, acos, asin, $\sqrt{x^2+y^2}$, x^n, $x^{1/n}$, $\sin(\pi(\cdot))$, $\cos(\pi(\cdot))$, atan/π, sinh, cosh, asinh, acosh, tanh, atanh.

La situation est autrement plus délicate lorsqu'il s'agit d'implémenter au niveau des flottants une fonction

$$f\ : (x_1, ..., x_n) \in \mathbb{R}^n \longrightarrow \mathbb{R}$$

et que l'on cherche à arrondir « correctement » $y = f(x_1, ..., x_n)$ pour $(x_1, ..., x_n)$ des flottants donnés. Une bonne approximation $\widehat{y} = y(1+\epsilon)$ de la fonction f permet

certes de trouver une grille de flottants avoisinant y, mais ne permet en général pas de décider quel flottant dans cette grille réalise un des cinq arrondis corrects (au sens de la norme `IEEE 754`) de y. Un test fait sur l'approximation $\widehat{y}$ ne saurait en effet induire de conclusion relativement à ce qui aurait du se passer si le test avait été conduit, comme il aurait du l'être, avec y en place de $\widehat{y}$.

Signalons les deux lemmes suivants (faciles à vérifier et souvent utiles comme « garde-fous », par exemple pour valider la preuve d'un postulat mathématique à l'aide d'une machine [4]).

LEMME 1.3 (lemme de Sterbenz, 1974). *Si x et y sont deux flottants tels que l'on ait $x/2 < y < 2x$, alors $x \ominus y = x - y$.*

LEMME 1.4 (erreur d'arrondi sur l'addition). *Si x et y sont des flottants, l'erreur d'arrondi $(x + y) - (x \oplus y)$ est un flottant et l'on a*

$$(x + y) - (x \oplus y) = y \ominus \big((x \oplus y) \ominus x\big). \tag{1.2}$$

Il en est de même pour $(x \times y) - (x \otimes y)$, mais la formule remplaçant (1.2) dans ce cas est plus complexe.

4. La preuve en 1998 par T. Hales de la conjecture de Kepler (1611) stipulant que densité de l'empilement cubique à faces centrées ($\pi/\sqrt{18} \simeq .74$) maximise la densité d'un empilement de sphères égales en est un exemple instructif.

CHAPITRE 2

Suites et séries, resommation, accélération de convergence

2.1. Séries entières génératrices

DÉFINITION 2.1 (série génératrice ordinaire et exponentielle, resommation de Borel). Soit $(a_n)_{n\in\mathbb{N}}$ une suite de nombres complexes. La série entière $(\sum_0^n a_k z^k)_{n\geq 0}$, série de fonctions d'une variable complexe $z \in \mathbb{C}$, est dite *série génératrice ordinaire* de la suite $(a_n)_{n\in\mathbb{N}}$. La série entière $(\sum_0^n a_k z^k/k!)_{n\geq 0}$ est dite *série génératrice exponentielle* de la suite $(a_n)_{n\geq 0}$, ou encore *resommée de Borel* de la série entière $(\sum_0^n a_k z^k)_{n\geq 0}$.

On rappelle ici le résultat majeur concernant les séries génératrices ordinaires.

PROPOSITION 2.1 (principe d'Abel). *Soit $(a_n)_{n\geq 0}$ une suite de nombres complexes. Si la série génératrice ordinaire $(\sum_0^n a_k z^k)_{n\geq 0}$ de la suite $(a_n)_{n\geq 0}$ converge en un point z_0 du plan complexe, elle converge normalement dans tout disque fermé $\overline{D(0,r)}$ avec $r < |z_0|$ et uniformément dans tout secteur angulaire conique*

$$K_\kappa(z_0) := \{z \in \mathbb{C}\,;\, |z| \leq |z_0| \;\&\; |z - z_0| \leq \kappa(|z_0| - |z|)\}\,,\ \kappa \geq 1,$$

en particulier sur le segment $[0, z_0] = K_1(z_0)$.

On rappelle aussi que l'on peut associer à la série génératrice ordinaire $(\sum_0^n a_k z^k)_{n\geq 0}$ son *rayon de convergence* défini par la *règle de Cauchy*[1] :

$$R = \frac{1}{\limsup\limits_{n\to+\infty} |a_n|^{1/n}} \in [0, \infty] \tag{2.1}$$

et encadré par la *règle de d'Alembert*[2] :

$$\frac{1}{\limsup\limits_{n\to+\infty} \frac{|a_{n+1}|}{|a_n|}} \leq R \leq \frac{1}{\liminf\limits_{n\to+\infty} \frac{|a_{n+1}|}{|a_n|}} \tag{2.2}$$

Ce rayon de convergence R est égal à la borne supérieure des nombres r positifs tels que la suite $(|a_n|r^n)_{n\geq 0}$ soit une suite bornée. Du point de vue de l'analyse numérique (qui est le point de vue sous tendant ce cours), on retiendra surtout que ceci implique le résultat suivant :

1. Nous retrouverons plusieurs fois le mathématicien français (en particulier analyse, on lui doit l'analyse moderne telle que nous en connaissons aujourd'hui la formalisation) Augustin Cauchy (1789-1857) dans ce cours.

2. Jean Le Rond d'Alembert (1717-1783), mathématicien français au siècle des lumières, fut l'un des pères de l'Encyclopédie.

PROPOSITION 2.2 (calcul numérique approché d'une série génératrice). *Soit $(a_n)_{n\geq 0}$ une suite de nombres complexes et R le rayon de convergence de la série génératrice associée (donné par la règle de Cauchy (2.1) ou tout au moins encadré par la règle de d'Alembert (2.2)). Pour tout z_0, r tels que $|z_0| < r < R$, pour tout $n \in \mathbb{N}$, on a*

$$\begin{aligned}\Big|\sum_{k=0}^{\infty} a_k z_0^k - \sum_{k=0}^{n} a_k z_0^k\Big| &\leq \sup_{k\geq 0}(|a_k| r^k) \times \sum_{k=n+1}^{\infty} \Big(\frac{|z_0|}{r}\Big)^k \\ &\leq \sup_{k\geq 0}(|a_k| r^k) \times \frac{|z_0|}{r-|z_0|} \times \Big(\frac{|z_0|}{r}\Big)^n \\ &\leq \sup_{k\geq 0}(|a_k| r^k) \times \frac{|z_0|}{r-|z_0|} \times \exp\Big(-n \log \frac{r}{|z_0|}\Big),\end{aligned}$$

autrement dit, l'erreur

$$\Big|\sum_{k=0}^{\infty} a_k z_0^k - \sum_{k=0}^{n} a_k z_0^k\Big|$$

tend exponentiellement vite vers 0 lorsque n tend vers l'infini.

REMARQUE 2.2 (sensibilisation aux notions de resommation et d'accélération de convergence). La conclusion de la Proposition 2.2 est en défaut lorsque $|z_0| = R$ et que la série génératrice $(\sum_0^n a_k z^k)_{n\geq 0}$ converge en z_0. Par exemple, si l'on prend $z_0 = 1$ dans la formule donnant la somme de la série génératrice

$$\Big(\sum_0^n \frac{(-1)^k z^{2k+1}}{2k+1}\Big)_{n\geq 0} \tag{2.3}$$

(il y a bien convergence en ce point du fait du critère des séries alternées), on obtient au mieux une estimation d'erreur [3] en

$$\Big|\mathrm{atan}(1) - \sum_{k=0}^{n} (-1)^k \frac{1}{2k+1}\Big| \leq \frac{1}{2(n+1)+1} = \frac{1}{2n+3}$$

et l'on peut aisément montrer qu'en fait cette erreur est équivalente à $1/(2n)$ lorsque n tend vers l'infini. Il s'agit là d'une convergence très lente et on est bien loin de la convergence exponentielle ! Pourtant une formule algébrique telle que

$$(5+i)^4 = 2(239+i)(1+i)$$

(à vérifier) nous assure que

$$\mathrm{atan}\,(1) = \frac{\pi}{4} = 4\,\mathrm{atan}\,(1/5) - \mathrm{atan}\,(1/239).$$

Comme $1/5 < 1$ et $1/239 < 1$ et que 1 est le rayon de convergence de la série génératrice (2.3) dont la somme donne dans le disque unité ouvert (et au point 1) la fonction

$$z \longmapsto \mathrm{atan}\,(z),$$

3. L'erreur entre la somme d'une série alternée et la somme des n premiers termes est majorée en module par le premier terme négligé, c'est-à-dire le $(n+1)$-ième.

on a bien convergence exponentielle de l'erreur dans la formule de John Machin (1706)

$$\frac{\pi}{4} = \lim_{n\to+\infty} \sum_{k=0}^{n} \frac{(-1)^k}{2k+1}\Big(4\times 5^{-(2k+1)} - 239^{-(2k+1)}\Big) \tag{2.4}$$

Une telle formule apparait comme une variante intelligente du procédé en deux temps (dit *taubérien*, ici il s'agit du *processus taubérien de Poisson*)

$$\operatorname{atan}(1) = \frac{\pi}{4} = \lim_{r\to 1^-} \operatorname{atan}(r) = \lim_{r\to 1^-} \Big(\lim_{n\to+\infty} \sum_{k=0}^{n} \frac{(-1)^k}{2k+1} r^{2k+1} \Big)$$

(notons que l'on a ici un problème délicat d'interversion de limites). Il s'agit là d'un mécanisme de *resommation*[4] qui sera à la base des procédés d'*accélération de convergence* (Richardson, Romberg) que nous retrouverons dans ce cours.

2.2. Série génératrice exponentielle, resommation de Borel

Si $(a_n)_{n\geq 0}$ est une suite de nombres complexes dont la série génératrice ordinaire $(\sum_0^n a_k z^k)_{n\geq 0}$ a un rayon de convergence $R > 0$, le procédé de Borel qui transforme la série génératrice ordinaire en la série génératrice exponentielle $(\sum_0^n a_k z^k/k!)_{n\geq 0}$ élargit à tout le plan complexe le domaine de convergence. On a en effet la :

PROPOSITION 2.3 (resommation de Borel). *Soit $(a_n)_{n\geq 0}$ est une suite de nombres complexes dont la série génératrice ordinaire $(\sum_0^n a_k z^k)_{n\geq 0}$ a un rayon de convergence $R > 0$. La série génératrice exponentielle $(\sum_0^n a_k z^k/k!)_{n\geq 0}$ a alors pour rayon de convergence $R = +\infty$. De plus, pour tout $r \in]0, R[$, il existe une constante $C(r)$ telle que*

$$\forall\, z\in\mathbb{C}\,,\ \Big|\sum_{k=0}^{\infty} \frac{a_k}{k!} z^k\Big| \leq \sum_{k=0}^{\infty} \frac{|a_k|}{k!}\,|z|^k \leq C(r)\exp\Big(\frac{|z|}{r}\Big). \tag{2.5}$$

Réciproquement, si la série entière $(\sum_0^n b_k z^k)_{n\geq 0}$ a un rayon de convergence égal à $+\infty$ et s'il existe des constantes C_0 et $r_0 > 0$ telles que

$$\forall\, z\in\mathbb{C}\,,\ \Big|\sum_{k=0}^{\infty} b_k z^k\Big| \leq C_0\exp\Big(\frac{|z|}{r_0}\Big)\,,$$

la série génératrice ordinaire de la suite $(b_n n!)_{n\geq 0}$ a un rayon de convergence R au moins égal à r_0.

DÉMONSTRATION. La preuve du premier point est facile. Si $r \in]0, R[$, on a $|a_n| \leq C(r) r^{-n}$ pour tout $n \geq 0$, avec $C(r) \geq 0$. On en déduit

$$\sum_{k=0}^{\infty} |a_k| \frac{|z|^k}{k!} \leq C(r) \sum_{k=0}^{\infty} \frac{(|z|/r)^k}{k!} = C(r)\exp(|z|/r),$$

d'où le résultat de l'implication directe. Pour la réciproque, on utilise par exemple la formule de Plancherel pour remarquer que, pour tout $r > 0$,

$$\sum_{k=0}^{\infty} |b_k|^2 r^{2k} = \frac{1}{2\pi}\int_0^{2\pi} \Big|\sum_{k=0}^{\infty} b_k r^k e^{ik\theta}\Big|^2 d\theta \leq C_0^2\exp(2r/r_0).$$

4. En un certain sens, nous avons « resommé » intelligemment en introduisant un mécanisme de pondération la série numérique $\sum_k (-1)^{2k+1}/(2k+1)$ convergeant trop lentement vers $\pi/4$.

Il en résulte

$$\begin{aligned} |b_k| \leq C_0 \min_{r>0} \exp(r/r_0 - k\log r) &= C_0 \Big[\exp(r/r_0 - k\log r) \Big]_{|r=kr_0} \\ &= C_0 \exp(k - k\log k - k\log r_0) \\ &= C_0 (e/k)^k r_0^{-k}. \end{aligned}$$

Si l'on utilise la *formule de Stirling*

$$k! \simeq \sqrt{2\pi k} \left(\frac{k}{e}\right)^k, \tag{2.6}$$

on constate que pour tout $r < r_0$, la suite $(|b_n|n!r^n)_{n\geq 0}$ tend vers 0 à l'infini et est donc bornée. Le secnd point de la proposition en résulte. □

Le ≪ retour ≫ de la série génératrice exponentielle à la série génératrice ordinaire se matérialise aussi par le biais d'une transformation que nous retrouverons ultérieurement, la *transformation de Laplace*.

PROPOSITION 2.4 (inversion de la resommation de Borel et transformée de Laplace). *Soit $(a_n)_{n\geq 0}$ une suite de nombres complexes dont la série génératrice ordinaire a un rayon de convergence $R > 0$. On note $F_{\rm ord} : D(0,R) \to \mathbb{C}$ la somme de sa série génératrice ordinaire et $F_{\rm exp} : \mathbb{C} \to \mathbb{C}$ la somme de sa série génératrice exponentielle. Alors, pour tout nombre complexe p tel que $\mathrm{Re}\, p > R$, l'intégrale*

$$\mathcal{L}[F_{\rm exp}](p) = \int_{[0,\infty[} F_{\rm exp}(t)\, e^{-pt}\, dt$$

définit une fonction analytique dans le demi-plan $\mathrm{Re}\, p > R$, dite transformée de Laplace de la restriction de $F_{\rm exp}$ à $[0,+\infty[$. Cette fonction analytique se prolonge dans $\{p \in \mathbb{C}\,;\, |p| > R\}$ en la fonction analytique

$$p \in \{|p| > R\} \longmapsto \frac{1}{p}\, F_{\rm ord}(1/p).$$

DÉMONSTRATION. Pour tout $\epsilon > 0$, il existe (Proposition 2.3, inégalité (2.5)) une constante C_ϵ telle que, pour tout $t \in [0,\infty[$,

$$\sum_{k=0}^{\infty} |a_k| \frac{t^k}{k!} \leq e^{(R+\epsilon)t}.$$

Par le théorème de convergence dominée de Lebesgue, on peut donc affirmer, puisque

$$\int_0^\infty \Big(\sum_{k=0}^{\infty} \frac{|a_k|}{k!} t^k \Big) |e^{-pt}|\, dt \leq C_\epsilon \int_0^\infty e^{(R+\epsilon-\mathrm{Re}\, p)\, t}\, dt < +\infty$$

si $\mathrm{Re}\, p > R + 2\epsilon$, que

$$\begin{aligned} \mathcal{L}[F_{\rm exp}](p) &= \int_0^{+\infty} \Big(\sum_{k=0}^{\infty} \frac{a_k}{p!} t^k \Big) e^{-pt}\, dt = \sum_{k=0}^{\infty} \frac{a_k}{k!} \int_0^\infty t^k e^{-pt}\, dt \\ &= \sum_{k=0}^{\infty} \frac{a_k}{k!} \frac{1}{p^{k+1}} \int_0^\infty u^k e^{-u}\, du = \sum_{k=0}^{\infty} \frac{a_k}{p^{k+1}} = \frac{1}{p}\, F_{\rm ord}(1/p). \end{aligned}$$

La fonction

$$p \in \{|p| > R\} \longmapsto \frac{1}{p}\, F_{\rm ord}(1/p)$$

(qui est analytique dans $\{|p| > R\}$) est le prolongement analytique de la fonction $\mathcal{L}[F_{\text{ext}}]$ à tout l'extérieur du disque fermé $\overline{D(0,R)}$. □

2.3. Produit de Cauchy et séries génératrices ordinaires

DÉFINITION 2.3 (produit de Cauchy). Si $(u_n)_{n\geq 0}$ et $(v_n)_{n\geq 0}$ sont deux suites de nombres complexes, on définit le produit de Cauchy des séries $\sum_n u_n$ et $\sum_n v_n$ comme la série $\sum_n w_n$ de terme général

$$w_n := \sum_{k=0}^{n} u_k v_{n-k}, \quad n \in \mathbb{N}\,.$$

REMARQUE 2.4. Au contraire du produit de Hadamard qui au couple de suites $(u_n)_{n\geq 0}$ et $(v_n)_{n\geq 0}$ associe la suite $(u_n v_n)_{n\geq 0}$, il est très important de penser le produit de Cauchy (dont on verra plus tard l'importance au niveau opérationnel) comme une opération sur les séries, c'est-à-dire les « suites cumulées ». C'est d'ailleurs ce dont rend compte la Proposition 2.5 suivante.

PROPOSITION 2.5 (produit de Cauchy et séries génératrices ordinaires). *Si $(u_n)_{n\geq 0}$ et $(v_n)_{n\geq 0}$ sont deux suites de nombres complexes, de séries génératrices ordinaires ayant pour somme respectivement F_{ord} et G_{ord} (avec rayons de convergence respectifs R_u et R_v), la série génératrice ordinaire de la suite $(w_n)_{n\geq 0}$ (où w_n désigne le terme général du produit de Cauchy des deux séries $\sum_n u_n$ et $\sum_n v_n$), a pour rayon de convergence $R_w = \inf(R_u, R_v)$ et l'on a*

$$\forall\, z \in D(0,R_w), \quad H(z) := \sum_{k=0}^{\infty} w_k z^k = \sum_{k=0}^{\infty} \Big(\sum_{l=0}^{k} u_l v_{k-l} \Big) z^k = F_{\text{ord}}(z) \times G_{\text{ord}}(z).$$

REMARQUE 2.5 (un résultat de Mertens). Si $R_u > R_v$ et que la série génératrice ordinaire de $(v_n)_{n\geq 0}$ converge en un point z_0 tel que $|z_0| = R_v$, alors la série génératrice ordinaire de $(w_n)_{n\geq 0}$ converge aussi au point z_0 et l'on a

$$H(z_0) = \sum_{k=0}^{\infty} w_k z_0^k = F_{\text{ord}}(z_0) \times G_{\text{ord}}(z_0).$$

Ce résultat s'avère toutefois en général en défaut lorsque $R_u = R_v$ et que la série génératrice ordinaire de la suite $(u_n)_{n\geq 0}$ n'est pas absolument convergente au point z_0 : pour s'en convaincre, prendre par exemple $(u_n)_{n\geq 0} = (v_n)_{n\geq 0}$ avec $u_n = (-1)^n/\sqrt{n+1}$; les séries génératrices ordinaires de ces deux suites convergent en $z_0 = 1$ (mais pas absolument !), tandis celle de la suite $(w_n)_{n\geq 0}$ diverge en ce point (on pourra faire l'exercice).

2.4. Convolution discrète (conv), transformée de Fourier discrète (dft)

Une importante opération algébrique soutend le calcul du produit de Cauchy.

DÉFINITION 2.6 (convolution discrète de deux suites *causales*[5]). La *convolée discrète*[6] des deux suites $(u_n)_{n\geq 0}$ et $(v_n)_{n\geq 0}$ est par définition la suite de terme

5. Nous parlons ici de suites causales car ultérieurement nous étendrons cette opération au cadre plus naturel des suites $(u_n)_{n\in\mathbb{Z}}$ indexées par $\mathbb{Z}$ ($\mathbb{Z}$ est équipé d'une structure de groupe, ce qui n'est pas le cas de $\mathbb{N}$). Par « causale », il faut entendre ici que $u_k = 0$ pour tout $k < 0$.

6. On dit aussi parfois *convoluée discrète*.

général

$$w_n = \sum_{k=0}^{n} u_k v_{n-k} = \sum_{k=0}^{n} u_{n-k} v_k, \quad n \in \mathbb{N}\,.$$

REMARQUE 2.7. Sous un environnement tel que `MATLAB` ou `Scilab`, la convolution d'une suite discrète $[u(0), ..., u(N_1-1)]$ (présentée sous forme de vecteur ligne) avec une suite $[v(0), ..., v(N_2-1)]$ (présentée aussi sous forme de vecteur ligne) est la suite de longueur $N_1 + N_2 - 1$ donnée par l'instruction :

```
w= conv (u,v);
```

REMARQUE 2.8. Une des raisons expliquant l'importance de la convolution discrète (dans le cas des suites finies) est qu'on la retrouve dans un contexte algébrique, celui des polynômes. Si $[u(0), ..., u(N_1-1)]$ et $[v(0), ..., v(N_2-1)]$ sont des suites finies (i.e de terme général nul pour n assez grand), les sommes de leurs séries génératrices ordinaires sont des fonctions polynomiales de la variable complexe z et la convolée des deux suites $(u_n)_{n\geq 0}$ et $(v_n)_{n\geq 0}$ représente (si tronquée à $N_1 + N_2 - 2$) d'après la Proposition 2.5 la suite des coefficients du polynôme de degré $(N_1 - 1) + (N_2 - 1) = N_1 + N_2 - 2$:

$$\Big(\sum_{k=0}^{N_1-1} u(k)\, X^k\Big) \times \Big(\sum_{k=0}^{N_2-1} v(k), X^k\Big) = \sum_{k=0}^{N_1+N_2-2} w(k)\, X^k. \tag{2.7}$$

C'est ainsi la multiplication des polynômes que traduit la convolution discrète.

Si l'on choisit $N \geq N_1 + N_2 - 1$, la relation (2.7) est équivalente à

$$\Big(\sum_{k=0}^{N_1-1} u(k)\, X^k\Big) \times \Big(\sum_{k=0}^{N_2-1} v(k)\, X^k\Big) \equiv \sum_{k=0}^{N_1+N_2-2} w(k)\, X^k \mod X^N - 1, \tag{2.8}$$

ou encore, puisque les N racines complexes du polynôme $X^N - 1$ sont les N racines complexes N-ième de l'unité $\exp(-2i\pi j/N)$, $j = 0, ..., N-1$, au système de N relations :

$$\Big(\sum_{k=0}^{N_1-1} u(k)\, e^{-\frac{2i\pi kj}{N}}\Big) \times \Big(\sum_{k=0}^{N_2-1} v(k)\, e^{-\frac{2i\pi kj}{N}}\Big) = \sum_{k=0}^{N_1+N_2-2} w(k)\, e^{-\frac{2i\pi kj}{N}}, \quad j = 0, ..., N-1. \tag{2.9}$$

Posons $W_N := \exp(-2i\pi/N)$ et remarquons maintenant la propriété algébrique suivante :

LEMME 2.9 (matrice de transformation de Fourier discrète [7] `dft`). *Soit $N \in \mathbb{N}^*$ et* `dft` (N) *la matrice carrée symétrique à coefficients complexes de terme général W_N^{kj}, k indice de ligne, j indice de colonne. La matrice* `dft` (N) *est inversible, d'inverse*

$$[\texttt{dft}\,(N)]^{-1} = \frac{1}{N}\,\overline{\texttt{dft}\,(N)}. \tag{2.10}$$

DÉMONSTRATION. Il suffit de remarquer que

$$\sum_{k=0}^{N-1} W_N^{jk} = \begin{cases} N & \text{si} \quad j = 0 \\ 0 & \text{si} \quad j \neq 0 \end{cases}$$

7. ≪ Pour *Discrete Fourier Transform* ≫.

car $X^N - 1 = (X-1)(1 + X + \cdots + X^{N-1})$. □

Si les vecteurs-ligne $[u(0), ..., u(N_1-1)]$, $[v(0), ..., v(N_2-1)]$, $[w(0), ..., w(N_1+N_2-2)]$ sont complétés par des zéros en des vecteurs ligne U, V, W de longueur N, puis transposés en des vecteurs colonne ${}^tU$, ${}^tV$ et ${}^tW$ de longueur N, on peut donc exprimer le système de relations (2.9) sous la forme

$$ {}^tW = \frac{1}{N}\,\overline{\mathtt{dft}\,(N)}\cdot\Big[\Big(\mathtt{dft}(N)\cdot{}^tU\Big).*\Big(\mathtt{dft}(N)\cdot{}^tV\Big)\Big] \tag{2.11}$$

où $.*$ désigne (comme par exemple dans les environnements `MATLAB` ou `Scilab`) la multiplication des vecteurs colonne de longueur N entrée par entrée.

2.5. Algorithme de transformation de Fourier rapide `fft`

Lorsque $N = 2^p$, avec $p \in \mathbb{N}^*$, on doit aux deux ingénieurs informaticiens américains James William Cooley and John Wilder Tukey (autour de 1965) la construction d'un algorithme permettant d'implémenter les opérations de multiplication matricielle

$$\mathtt{dft}(2^p)\cdot{}^tU,\quad \Big(\text{resp.}\quad \frac{1}{2^p}\,\overline{\mathtt{dft}(2^p)}\cdot{}^t\widehat{W}\Big),$$

où ${}^tU$ et ${}^t\widehat{W}$ sont deux vecteurs colonne de longueur 2^p donnés, avec $p2^{p-1}$ (au lieu de $(2^p)^2 = 2^{2p}$) multiplications (en fait seulement $(p-1)2^{p-1}$ si l'on prend en compte que 2^{p-1} de ces multiplications sont des multiplications par -1, que l'on peut donc considérer comme des additions au niveau de la complexité). Nous énonçons ici ce résultat majeur, moteur de ce qui allait être la *révolution numérique* des années 1968-1970.

Theorème 2.10 (algorithme de Cooley-Tukey (environ 1965)). *Lorsque $N = 2^p$, $p \in \mathbb{N}^*$, il existe un algorithme consommant seulement $p2^{p-1}$ multiplications (parmi lesquelles 2^{p-1} sont des multiplications par -1) permettant la multiplication matricielle*

$$\mathtt{dft}(2^p)\cdot{}^tU,\quad \Big(\text{resp.}\quad \frac{1}{2^p}\,\overline{\mathtt{dft}(2^p)}\cdot{}^t\widehat{W}\Big),$$

lorsque ${}^tU$ et ${}^t\widehat{W}$ sont deux vecteurs colonne donnés de longueur 2^p. Cet algorithme est appelé « Algorithme de Transformation de Fourier Rapide », ou plus communément $\mathtt{fft}(2^p)$ *pour « Fast Fourier Transform » (respectivement « Algorithme de Transformation de Fourier Rapide Inverse », ou plus communément* $\mathtt{ifft}(2^p)$ *pour « Inverse Fast Fourier Transform »).*

Remarque 2.11. Sous l'environnement `MATLAB` ou `Scilab`, ces algorithmes s'implémentent *via* les routines

```
>> (hatU)' = fft(U',N);
>> W' = ifft ((hatW)',N);
```

Démonstration. La clef de l'algorithme de Cooley-Tukey consiste à profiter du fait que l'on a

$$\mathtt{dft}(2) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

(l'action de cette matrice, dite « action papillon », est donc très simple) et à enchainer, pour calculer une prise de $\mathtt{fft}(2^k)$, deux prises de $\mathtt{fft}(2^{k-1})$ avec 2^{k-1} prises

de `dft`(2). Il est commode de visualiser cette idée (sous-tendant la récursivité) grâce au diagramme suivant (où $W_N := \exp(-2i\pi/N)$).

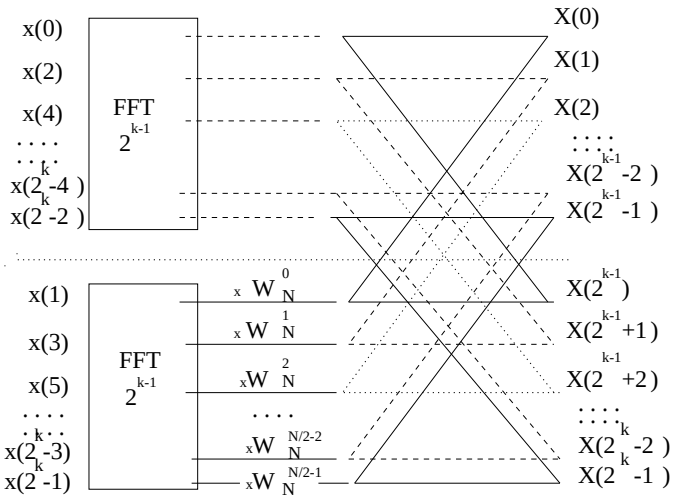

FIGURE 1. *Algorithme de Cooley Tukey* $N/2 = 2^{k-1} \to N = 2^k$

L'algorithme récursif à implémenter pour calculer par exemple $\texttt{dft}(2^p) \cdot X$ est présenté comme l'algorithme 1 ci-dessous. On remarque que cet algorithme implique à chaque étape (lignes 5 et 6) une réorganisation des entrées (réalisée par un renversement de l'un des bits des indices des entrées, $0 \to 1$, $1 \to 0$). Il est aisé de constater que cet algorithme consomme

$$\frac{2^p}{2} + 2\frac{2^p}{4} + \cdots + 2^{p-1}\frac{2^p}{2^p} = p\frac{2^p}{2} = p \times 2^{p-1}$$

multiplications, dont 2^{p-1} sont en fait des multiplications par $e^{-2i\pi/2} = -1$. L'algorithme $\texttt{ifft}(2^p)$ est en tous points similaire : il suffit de remplacer w par $\overline{w}$ et de conclure avec une division par N. □

Algorithme 1 `fft` $(X, Y, 2^p, w)$

1: $w \Leftarrow e^{-2i\pi/2^p}$
2: **si** $p = 0$ **alors**
3: $\quad Y(0) \Leftarrow X(0)$
4: **sinon**
5: $\quad$ `fft` $([X(0), ..., X(2^p - 2)], [U(0), ..., U(2^{p-1})], 2^{p-1}, w^2)$
6: $\quad$ `fft` $([X(1), ..., X(2^p - 1)], [V(0), ..., V(2^{p-1})], 2^{p-1}, w^2)$
7: $\quad$ **pour** $k = 0$ **jusqu'à** $2^p - 1$ **faire**
8: $\quad\quad Y(k) \Leftarrow U(k \bmod 2^{p-1}) + w^k\, V(k \bmod 2^{p-1})$
9: $\quad$ **fin pour**
10: **fin si**

2.6. Calculs ≪ accélérés ≫ de limites de suites ou de séries numériques

2.6.1. Deux procédés d'accélération sur un modèle simple.

A. *Le modèle de L.F. Richardson sur un exemple*

On considère dans cette sous-section une suite numérique $(u_n)_{n\geq 0} = (u_n^{(0)})_{n\geq 0}$ de nombres complexes dont on sait *a priori* qu'elle converge vers une limite l, la convergence étant de la forme

$$u_n - l = \lambda h^n(1+\epsilon_n), \quad \lambda \in \mathbb{C}^*,\ h \in D(0,1)\setminus\{0\}, \quad \text{avec} \quad \lim_{n\to+\infty} \epsilon_n = 0. \tag{2.12}$$

La suite de terme général

$$u_n^{(1)} := \frac{u_{n+1}^{(0)} - h u_n^{(0)}}{1-h}$$

converge également vers l (compte tenu de l'hypothèse (2.12)). Mais on remarque que

$$\begin{aligned} u_n^{(1)} - l &= \frac{u_{n+1}^{(0)} - h u_n^{(0)}}{1-h} - l \\ &= \frac{1}{1-h}\Big(l + \lambda h^{n+1}(1+\epsilon_{n+1}) - hl - \lambda h^{n+1}(1+\epsilon_n) - l(1-h)\Big) \\ &= \frac{\lambda}{1-h}\, h^{n+1}(\epsilon_{n+1} - h\epsilon_n). \end{aligned}$$

Tandis que $|u_n^{(0)} - l| = O(h^n)$, on voit que $|\widetilde{u}_n^{(1)} - l| = o(h^n)$, ce qui prouve que la convergence de $u_n - l$ vers 0 se trouve bien ≪ accélérée ≫ si l'on opère la substitution $u_n = u_n^{(0)} \to u_n^{(1)}$. Ce premier modèle d'accération de convergence illustre ce qui sera, dans un contexte plus général, le procédé d'≪ accélération ≫ (ou encore, on verra pourquoi, d'≪ extrapolation ≫) de L. F. Richardson [8]. Si l'on pose $x_k = h^k$ pour tout $k \in \mathbb{N}$, on remarque que l'on peut écrire

$$u_n^{(1)} = \frac{u_{n+1}^{(0)} x_n - u_n^{(0)} x_{n+1}}{x_n - x_{n+1}},$$

autrement dit le terme général de la suite $(u_n^{(1)})_{n\geq 0}$ est obtenu en construisant le polynôme d'interpolation de Lagrange [9]

$$\text{Lagrange}\Big(x_n, x_{n+1}\,;\, u_n^{(0)}, u_{n+1}^{(0)}\Big)[X] = u_n^{(0)} + (X - x_n)\frac{u_{n+1}^{(0)} - u_n^{(0)}}{x_{n+1} - x_n}$$

interpolant les valeurs $u_n^{(0)}$ et $u_{n+1}^{(0)}$ respectivement aux points x_n et x_{n+1}, puis en formant

$$u_n^{(1)} := \text{Lagrange}\Big(x_n, x_{n+1}\,;\, u_n^{(0)}, u_{n+1}^{(0)}\Big)[0]. \tag{2.13}$$

C'est ce processus que nous itérerons à partir d'une suite de référence $(x_n)_{n\geq 0}$ (convergeant vers 0 et composée de nombres complexes non nuls distincts) dans la

8. Les travaux du physicien et mathématicien anglais Lewis Fry Richardson (1881-1953) ont été pour une grande part tournés vers les prévisions météorologistes ; c'est dans cette optique qu'a surgi la technique d'extrapolation (et d'accélération de convergence) que nous mentionnons ici.

9. On se reportera au cours de MHT304 (voir [**Y1**]) pour la définition du polynôme d'interpolation de Lagrange.

sous-section suivante. C'est aussi la présentation (2.13) qui explique le qualificatif de « méthode d'extrapolation » pour dénommer le procédé d'accélération ainsi décrit dans ce contexte particulier : la valeur $u_n^{(1)}$ est obtenue en effet en « extrapolant » la fonction digitale $x_n \mapsto u_n^{(0)}$, $x_{n+1} \mapsto u_{n+1}^{(0)}$ de l'ensemble $\{x_n, x_{n+1}\}$ jusqu'à la valeur $\xi = 0$, ce *via* le procédé d'interpolation de Lagrange.

B. *Le Δ^2-algorithme de A.C. Aitken sur un exemple*

Reprenons la suite $(u_n)_{n\geq 0} = (\widetilde{u}_n^{(0)})_{n\geq 0}$ satisfaisant (2.12). On a, pour tout $n \geq 0$,

$$\begin{aligned} \Delta\widetilde{u}_n^{(0)} := \widetilde{u}_n^{(0)} - \widetilde{u}_{n+1}^{(0)} &= -\lambda h^{n+1}(1+\epsilon_{n+1}) + \lambda h^n(1+\epsilon_n) \\ &= \lambda h^n(1-h+h\epsilon_{n+1}-\epsilon_n) \sim \lambda(1-h)h^n \\ &\qquad \text{quand} \quad n \to +\infty. \end{aligned}$$

Ceci assure, pour n assez grand ($n \geq N$), que $\Delta\widetilde{u}_n^{(0)} \neq 0$ et que l'on peut alors définir

$$\widetilde{v}_n := \frac{\Delta\widetilde{u}_{n+1}^{(0)}}{\Delta\widetilde{u}_n^{(0)}}.$$

De plus

$$1 - \widetilde{v}_n = \frac{2\widetilde{u}_{n+1}^{(0)} - \widetilde{u}_n^{(0)} - \widetilde{u}_{n+2}^{(0)}}{\widetilde{u}_{n+1}^{(0)} - \widetilde{u}_n^{(0)}}.$$

Or

$$\begin{aligned} \frac{2\widetilde{u}_{n+1}^{(0)} - \widetilde{u}_n^{(0)} - \widetilde{u}_{n+2}^{(0)}}{\widetilde{u}_{n+1}^{(0)} - \widetilde{u}_n^{(0)}} &= \frac{\lambda h^n(2h-1-h^2+2h\epsilon_{n+1}-\epsilon_n-h^2\epsilon_{n+2})}{\lambda h^n(h-1+h\epsilon_{n+1}-\epsilon_n)} \\ &\sim 1-h \quad \text{quand} \quad n \to +\infty. \end{aligned}$$

Pour $n \geq \widetilde{N} \geq N$, on peut donc assurer $1 - \widetilde{v}_n \neq 0$. Ceci nous permet de définir simultanément

$$\begin{aligned} &\widetilde{v}_n := \frac{\Delta\widetilde{u}_{n+1}^{(0)}}{\Delta\widetilde{u}_n^{(0)}} \\ &\widetilde{u}_n^{(1)} := \frac{\widetilde{u}_{n+1}^{(0)} - \widetilde{v}_n\widetilde{u}_n^{(0)}}{1-\widetilde{v}_n} = \frac{\Delta\widetilde{u}_n^{(0)}\,\widetilde{u}_{n+1}^{(0)} - \Delta\widetilde{u}_{n+1}^{(0)}\,\widetilde{u}_n^{(0)}}{\Delta\widetilde{u}_n^{(0)} - \Delta\widetilde{u}_{n+1}^{(0)}}. \end{aligned} \tag{2.14}$$

Pour n tendant vers $+\infty$, on a

$$(\widetilde{u}_{n+1}^{(0)} - l) - h(\widetilde{u}_n^{(0)} - l) + (h - \widetilde{v}_n)(\widetilde{u}_n^{(0)} - l) = \lambda h^n(h(\epsilon_{n+1}-\epsilon_n) + (h-\widetilde{v}_n)(1+\epsilon_n)). \tag{2.15}$$

Ainsi, pour n tendant vers $+\infty$ (et strictement supérieur à $\widetilde{N}$),

$$\begin{aligned} |\widetilde{u}_n^{(1)} - l| &= \Big|\frac{\widetilde{u}_{n+1}^{(0)} - \widetilde{v}_n\widetilde{u}_n^{(0)} - l(1-\widetilde{v}_n)}{1-\widetilde{v}_n}\Big| \\ &= \Big|\frac{\widetilde{u}_{n+1}^{(0)} - l - \widetilde{v}_n(\widetilde{u}_n^{(0)} - l)}{1-\widetilde{v}_n}\Big| = o(|h|^n) = o(|\widetilde{u}_n^{(0)} - l|). \end{aligned}$$

Ici encore, la substitution $u_n = \widetilde{u}_n^{(0)} \to \widetilde{u}_n^{(1)}$ accélère le processus de convergence vers l. Cette démarche préfigure ici le procédé Δ^2 d'« accélération » (ou encore d'« extrapolation ») de A. C. Aitken [10].

10. Mathématicien néo-zélandais, Alexander Craig Aitken, 1895-1967, est aussi connu comme un calculateur mental « prodige ».

Voici une autre interprétation de cette construction, qui permet de la relier au procédé d'extrapolation de L.F. Richardson envisagé sur le même exemple ≪ jouet ≫ dans la sous-section précédente. Pour $n \geq \widetilde{N}$, on peut introduire le polynôme d'interpolation de Lagrange

$$\begin{aligned} &\text{Lagrange}\left(\widetilde{u}_{n+1}^{(0)} - u_n^{(0)}, u_{n+2}^{(0)} - u_{n+1}^{(0)}\,;\, \widetilde{u}_n^{(0)}, \widetilde{u}_{n+1}^{(0)}\right)[X] \\ &= \widetilde{u}_n^{(0)} + (X - \widetilde{u}_n^{(0)})\,\frac{\widetilde{u}_{n+1}^{(0)} - \widetilde{u}_n^{(0)}}{\widetilde{u}_{n+2}^{(0)} - 2\widetilde{u}_{n+1}^{(0)} + \widetilde{u}_n^{(0)}} \end{aligned}$$

et calculer sa valeur en $X = 0$, ce qui donne exactement

$$\begin{aligned} &\text{Lagrange}\left(\widetilde{u}_{n+1}^{(0)} - \widetilde{u}_n^{(0)}, \widetilde{u}_{n+2}^{(0)} - \widetilde{u}_{n+1}^{(0)}\,;\, \widetilde{u}_n^{(0)}, \widetilde{u}_{n+1}^{(0)}\right)[0] = \frac{\widetilde{u}_{n+1}^{(0)} - \widetilde{v}_n \widetilde{u}_n^{(0)}}{1 - \widetilde{v}_n} \\ &= \frac{\Delta\widetilde{u}_n^{(0)}\,\widetilde{u}_{n+1}^{(0)} - \Delta\widetilde{u}_{n+1}^{(0)}\,\widetilde{u}_n^{(0)}}{\Delta\widetilde{u}_n^{(0)} - \Delta\widetilde{u}_{n+1}^{(0)}} = \widetilde{u}_n^{(1)}. \end{aligned} \tag{2.16}$$

Cette ré-écriture de $\widetilde{u}_n^{(1)}$ (couplée avec le fait que le calcul de $\widetilde{u}_n^{(1)}$ fasse apparaitre explicitement l'action de l'opérateur de différentiation discrète Δ^2, agissant sur la suite initiale $(u_n^{(0)})_{n\geq 0}$) justifie le qualificatif de ≪ méthode Δ^2 d'extrapolation d'Aitken ≫ pour cette méthode.

2.6.2. Le procédé d'extrapolation de Richardson et l'accélération de convergence. Le procédé d'extrapolation (et dans les bons cas d'accélération de convergence) de L. W. Richardson est construit sur le principe de l'interpolation de Lagrange [11].

Supposons que l'on dispose d'une suite $(u_n)_{n\geq 0} = (u_n^{(0)})_{n\geq 0}$ convergeant (mais *a priori* lentement) vers une limite l.

On se fixe une suite de référence $(x_n)_{n\geq 0}$ de nombres strictement positifs, tendant vers 0 lorsque n tend vers $+\infty$, et telle que

$$\frac{x_n}{x_{n+1}} \geq c > 1 \quad \forall\, n \in \mathbb{N}. \tag{2.17}$$

On définit un double tableau de suites $(u_n^{(k)})_{n\geq 0, k\geq 0}$ en posant, pour tout $n \in \mathbb{N}$, pour tout $k \in \mathbb{N}$,

$$\begin{aligned} u_n^{(k+1)} &:= \text{Lagrange}\,[x_n, x_{n+k+1}; u_n^{(k)}, u_{n+1}^{(k)}]\,(0) \\ &= \frac{x_n u_{n+1}^{(k)} - x_{n+k+1} u_n^{(k)}}{x_n - x_{n+k+1}}. \end{aligned} \tag{2.18}$$

Pour chaque $k = 0, 1, 2, ...$, on construit ainsi une nouvelle suite $(u_n^{(k)})_{n\geq 0}$, convergeant elle aussi vers la limite l du fait de la condition (2.17). Le terme général $u_n^{(k)}$ de la suite $(u_n^{(k)})_{n\geq 0}$ (lorsque $k \geq 1$) correspond de fait à

$$u_n^{(k)} = \text{Lagrange}\,[x_n, x_{n+1}, ..., x_{n+k}; u_n^{(0)}, u_{n+1}^{(0)}, ..., u_{n+k}^{(0)}]\,(0). \tag{2.19}$$

Ceci est une conséquence de Lemme d'Aitken (permettant le calcul *via* un algorithme récursif du polynôme d'interpolation de Lagrange) :

11. Mathématicien, mais aussi astronome et mécanicien, le franco-italien Joseph-Louis Lagrange (1736-1813) fut l'inventeur du calcul variationnel et l'un des pères de l'optimisation.

LEMME 2.12 (lemme d'Aitken). *Soient $\xi_0, ..., \xi_N$ $N+1$ nombres réels distincts et $\eta_0, ..., \eta_N$ $N+1$ nombres complexes. Si Q désigne le polynôme d'interpolation de Lagrange interpolant les valeurs $\eta_0, ..., \eta_{N-1}$ respectivement aux points $\xi_0, ..., \xi_{N-1}$, R le polynôme d'interpolation de Lagrange interpolant $\eta_1, ..., \eta_N$ respectivement aux points $\xi_1, ..., \xi_N$, on a, si P désigne le polynôme d'interpolation de Lagrange interpolant les valeurs $\eta_0, ..., \eta_N$ aux points $\xi_0, ..., \xi_N$:*

$$P(X) = \frac{(X-\xi_0)R(X) - (X-\xi_N)Q(X)}{\xi_N - \xi_0}$$

En particulier

$$P(0) = \frac{\xi_0 R(0) - \xi_N Q(0)}{\xi_0 - \xi_N}.$$

On remarque que l'on a, pour tout $n, k \in \mathbb{N}$,

$$\begin{aligned} \frac{u_n^{(k+1)} - l}{u_n^{(k)} - l} &= \frac{x_n(u_{n+1}^{(k)} - l) - x_{n+k+1}(u_n^{(k)} - l)}{(u_n^{(k)} - l)(x_n - x_{n+k+1})} \\ &= \frac{1}{1 - x_{n+k+1}/x_n} \times \Big(\frac{u_{n+1}^{(k)} - l}{u_n^{(k)} - l} - \frac{x_{n+k+1}}{x_n} \Big). \end{aligned}$$

Ceci implique, compte tenu de la condition (2.17), que pour que la suite $(u_n^{(k+1)})_{n\geq 0}$ converge vers l plus rapidement que la suite $(u_n^{(k)})_{n\geq 0}$, c'est-à-dire

$$\lim_{n\to 0} \frac{u_n^{(k+1)} - l}{u_n^{(k)} - l} = 0,$$

ce qui est avant tout ce qui nous intéresse ici, il faut et il suffit que l'on ait

$$\lim_{n\to+\infty} \frac{u_{n+1}^{(k)} - l}{u_n^{(k)} - l} = \lim_{n\to+\infty} \frac{x_{n+k+1}}{x_n}. \tag{2.20}$$

Il s'avère cependant que dans la pratique cette condition (2.20) est difficile à réaliser. Elle pré-suppose un « principe de formation » de la suite initiale $(u_n^{(0)})_{n\geq 0}$ qu'il convient de connaitre au moins approximativement pour choisir la suite $(x_n)_{n\geq 0}$ strictement décroissante vers 0 et satisfaisant en outre les contraintes (2.17) et (2.20).

2.6.3. Le ϵ-algorithme scalaire d'Aitken. Pour implémenter de manière itérative la démarche de l'algorithme d'extrapolation (et éventuellement dans les bons cas, d'accélération de convergence) d'Aitken présentée (au premier cran) en **B**, on introduit (suivant une démarche initiée par P. Wynn in 1956) un tableau de suites à double entrée (k en entrée horizontale, cette fois indexé de -1 à l'infini, et n en entrée verticale).

La suite $(\epsilon_n^{(-1)})_{n\geq 0}$ est la suite indentiquement nulle, la suite $(\epsilon_n^{(0)})_{n\geq 0}$ est la suite d'étude $(u_n)_{n\geq 0}$ (dont on sait *a priori* qu'elle converge vers l). Les suites colonne suivantes ($k = 1, 2, ...$) sont construites suivant la règle de récurrence à deux pas :

$$\epsilon_n^{(k)} = \epsilon_{n+1}^{(k-2)} + \frac{1}{\epsilon_{n+1}^{(k-1)} - \epsilon_n^{(k-1)}}, \quad k \geq 2. \tag{2.21}$$

On remarque que l'on a

$$\forall\, n \in \mathbb{N},\ \epsilon_n^{(2)} = \widetilde{u}_n^{(1)},$$

où $(\widetilde{u}_n^{(1)})_{n\geq 0}$ est la suite déduite de la suite initiale $(u_n)_{n\geq 0} = (\widetilde{u}_n^{(0)})_{n\geq 0}$ dans le processus en deux temps indiqué en (2.14).

L'implémentation de cet algorithme s'effectue sous une forme dite ≪ en losange ≫ (et non plus ≪ en triangle ≫ comme c'était le cas pour l'algorithme d'extrapolation de Richardson basé sur le calcul inductif (2.18)). La ≪ maille ≫ de calcul se présente en effet sous la forme :

$$\begin{array}{ccccc} & & \epsilon_n^{(k-1)} & & \\ & \nearrow & & \searrow & \\ \epsilon_{n+1}^{(k-2)} & & \xrightarrow{+} & & \epsilon_n^{(k)} = \dfrac{1}{\epsilon_{n+1}^{(k-1)} - \epsilon_n^{(k-1)}} + \epsilon_n^{(k-2)} \\ & \searrow & & \nearrow & \\ & & \epsilon_{n+1}^{(k-1)} & & \end{array}$$

Dans le tableau triangulaire ainsi formé (et dont les colonnes sont numérotées à partir de $k = -1$: $k = -1, 0, 1, 2, ...$), les suites intéressantes à retenir sont celles qui sont indexées par les colonnes d'indice $k = 0$, $k = 2$,..., donc par les colonnes d'indice pair. Les autres colonnes (celles d'indice impair $k = -1, 1, 3, ...$) ne sont appelées qu'à jouer un rôle de colonnes auxiliaires dans les calculs.

La règle de calcul qui soutend le fait que cette démarche puisse conduire à une méthode d'accélération de convergence est la suivante (on en laisse la démonstration en exercice) :

PROPOSITION 2.6 (Clause d'accélération de convergence pour le Δ^2-algorithme d'Aitken). *Si la suite $(\epsilon_n^{(2k)})_{n\geq 0}$ converge vers une limite l et si*

$$\lim_{n\to+\infty} \frac{\epsilon_{n+1}^{(2k)} - l}{\epsilon_n^{(2k)} - l} = \lim_{n\to+\infty} \frac{\Delta\epsilon_{n+1}^{(2k)}}{\Delta\epsilon_n^{(2k)}} = h \neq 1\,, \tag{2.22}$$

alors la suite $(\epsilon_n^{(2(k+1)})_{n\geq 0}$ tend aussi vers l et on a

$$|\epsilon_n^{(2(k+1))} - l| = o(|\epsilon_n^{(2k)} - l|),$$

autrement dit la convergence vers l a bien été accélérée si l'on utilise comme suite d'approche de l la suite $(\epsilon_n^{(2(k+1))})_{n\geq 0}$ (i.e la suite figurant en $2(k+1)$-ième colonne du tableau) à la place de la suite $(\epsilon_n^{(2k)})_{n\geq 0}$ (qui elle figure en $2k$-ième colonne du tableau).

REMARQUE 2.13. Bien que nous disposions de la Proposition 2.6, il faut souligner qu'il existe peu de résultats théoriques généraux (en termes de gain en accélération de convergence) sur les applications répétées du Δ^2 algorithme d'extrapolation d'Aitken dans le ϵ-algorithme. La Proposition 2.6 (qui d'ailleurs ne donne qu'une condition suffisante pour qu'il y ait bien accélération à un cran $2(k+1)$-donné) s'avère tout aussi délicate à manier du point de vue pratique que ne l'est la condition (2.20) dans le processus d'extrapolation de Richardson.

Une remarque importante concernant la construction de la suite $(\epsilon_n^{(2(k+1))})_{n\geq 0}$ à partir de la suite $(\epsilon_n^{(2k)})_{n\geq 0}$ est la suivante : si l'on dispose de cette dernière suite,

on ajuste, pour chaque n fixé, trois nombres a, b, c pour que

$$\begin{aligned} \epsilon_n^{(2k)} &= a + bc^n \\ \epsilon_{n+1}^{(2k)} &= a + bc^{n+1} \\ \epsilon_{n+2}^{(2k)} &= a + bc^{n+2}, \end{aligned} \tag{2.23}$$

puis, une fois cet ajustement fait, on pose

$$\epsilon_n^{(2(k+1))} = a.$$

Ceci justifie une fois encore que la démarche sous-tendant le Δ^2-algorithme d'Aitken (et donc le ϵ-algorithme scalaire) soit bien une méthode d'extrapolation.

2.6.4. Versions vectorielles et matricielles du ϵ-algorithme. La transcription du ϵ-algorithme au cadre vectoriel ne pose pas de problème majeur. Les entrées $\epsilon_n^{(k)}$ du tableau à deux indices introduit dans la sous-section précédente étant des vecteurs colonne $\vec{\epsilon}_n^{(k)}$ de $\mathbb{R}^p$ ou $\mathbb{C}^p$, on adopte (à la place de la relation scalaire (2.21)) comme formule inductive :

$$\vec{\epsilon}_n^{(k)} = \vec{\epsilon}_{n+1}^{(k-2)} + \texttt{ones(1,p)}\,./(\vec{\epsilon}_{n+1}^{(k-1)} - \vec{\epsilon}_n^{(k-1)}).$$

Si les entrées $\epsilon_n^{(k)}$ sont toutes des matrices $p \times p$ à entrées réelles ou complexes, la relation récurrente (2.21) est naturellement à remplacer par

$$\epsilon_n^{(k)} = \epsilon_{n+1}^{(k-2)} + \left(\epsilon_{n+1}^{(k-1)} - \epsilon_n^{(k-1)}\right)^{-1},$$

tant que bien sûr la matrice carrée $\epsilon_{n+1}^{(k-1)} - \epsilon_n^{(k-1)}$ se trouve être inversible. Dans ce cadre vectoriel ou matriciel, le ϵ-algorithme peut par exemple être utilisé pour accélérer la convergence des algorithmes itératifs du type Jacobi ou Gauss-Seidel.

CHAPITRE 3

Algorithmique numérique et calcul différentiel

3.1. La formule de Taylor : du continu au discret

3.1.1. Le cas 1D. La formule de Taylor [1] s'avère du point de vue théorique une formule capitale pour approcher les fonctions d'une variable réelle t (ce sera souvent la variable temporelle, d'où cette notation) par des fonctions polynomiales (seules fonctions, avec les fractions rationnelles, codables en machine sous la forme de listes de coefficients). On en retiendra trois formes, dont une seule se trouve être totalement explicite. On rappelle ici ces trois formules dans le cadre $1D$. On renvoie au cours d'Analyse de L1 (MHT202) pour les preuves de ces résultats (voir aussi [**MathL2**], chapitre 14).

La première de ces trois formules de Taylor 1D est une formule ≪ locale ≫ dans laquelle le reste n'est pas explicité.

PROPOSITION 3.1 (formule de Taylor-Young [2]). *Si f est une fonction à valeurs réelles ou complexes définie au voisinage d'un point t_0 de $\mathbb{R}$ et dérivable à l'ordre $m \in \mathbb{N}$ au point t_0 (f est dérivable au voisinage de t_0, f',...,$f^{(m-1)}$ aussi et la dérivée de $f^{(m-1)}$ en t_0 existe), alors*

$$f(t_0 + h) = \sum_{k=0}^{m} \frac{f^{(k)}(t_0)}{k!}\, h^k + o(|h|^m) \tag{3.1}$$

lorsque h tend vers 0.

REMARQUE 3.1. Si l'on songe (par exemple) que $10^{10^5}|h|^{m+1}$ est tout de même un $o(|h|^m)$ (ce malgé la présence de l'énorme coefficient multiplicatif devant), on comprend le peu d'intérêt de la formule de Taylor-Young en ce qui concerne l'algorithmique numérique. Cette formule théorique certes intéressante, ne permet pas, de par le fait qu'elle ne soit pas quantifiable, de disposer d'un contrôle d'erreur *a priori* dans l'approximation des fonctions par la partie principale de leur développement de Taylor.

La seconde de ces trois formules de Taylor 1D est une formule non plus locale mais ≪ semi-locale ≫, c'est-à-dire en relation avec le comportement de f non plus au voisinage d'un point t_0 de $\mathbb{R}$, mais sur un segment temporel d'étude $[t_0 - H, t_0 + H]$. Sans être totalement explicite, elle autorise un contrôle d'erreur (entre f et son approximation polynomiale) bien utile lorsqu'il s'agira d'algorithmique numérique. Nous verrons d'ailleurs que c'est ce type de formule dont nous écrirons ultérieurement

1. On en doit l'introduction au scientifique anglais Brook Taylor (1685-1731).
2. Au nom de Taylor est associé celui de l'analyste anglais beaucoup plus récent William Henry Young (1863-1942).

une version « décentrée » (Proposition 3.4) en relation avec les *différences divisées*, substituts aux coefficients de Taylor en un point dans le cadre des mathématiques discrètes.

PROPOSITION 3.2 (formule de Taylor-Lagrange). *Si f est une fonction à valeurs réelles de classe C^m sur un segment $[t_0 - H, t_0 + H]$ de $\mathbb{R}$ (ceci signifie que f se prolonge en une fonction de classe C^m dans un intervalle ouvert $]t_0-H-\epsilon, t_0+H+\epsilon[$ pour un certain $\epsilon > 0$) telle que $f^{(m)}$ soit dérivable en tout point ξ de $]t_0-H, t_0+H[$, on a :*

$$\begin{aligned} &\forall\, h \in]-H, H[,\ \exists \xi_h \text{ entre } t_0 \text{ et } t_0 + h \quad t.q \\ &f(t_0 + h) = \sum_{k=0}^{m} \frac{f^{(k)}(t_0)}{k!}\, h^k + \frac{h^{m+1}}{(m+1)!} f^{(m+1)}(\xi_h). \end{aligned} \tag{3.2}$$

REMARQUE 3.2. Si f est à valeurs complexes (ou plus généralement vectorielles), la formulation (3.2) n'est plus valide et il faut se contenter d'une version « inégalité » :

$$\forall\, h \in]-H, H[, \quad \left\| f(t_0 + h) - \sum_{k=0}^{m} \frac{f^{(k)}(t_0)}{k!}\, h^k \right\| \leq \frac{|h|^{m+1}}{(m+1)!} \sup_{\xi \in]t_0, t_0+h[} \|f^{(m+1)}(\xi)\|. \tag{3.3}$$

Cette version *a priori* plus faible n'en est pas moins utile en algorithmique pour effectuer des estimations d'erreur.

La dernière des trois formules de Taylor rappelées dans ce cours est elle une formule totalement explicite, avatar de la formule d'intégration par parties. Elle est, ici encore, subordonnée à un segment $[t_0-H, t_0+H]$ donné de $\mathbb{R}$. On peut la considérer cette fois comme une formule globale, que l'on envisagera sous forme bilatérale (à gauche ou à droite de t_0). On notera que cette formule exacte requiert un cran de régularité de plus ($m+1$ au lieu de m) que les formules de Taylor-Young ou de Taylor-Lagrange.

PROPOSITION 3.3 (formule de Taylor avec reste intégral). *Si f est une fonction à valeurs réelles de classe C^{m+1} ($m \in \mathbb{N}$) sur un segment $[t_0 - H, t_0 + H]$ de $\mathbb{R}$ (ceci signifie que f se prolonge en une fonction de classe C^{m+1} dans un intervalle ouvert $]t_0 - H - \epsilon, t_0 + H + \epsilon[$ pour un certain $\epsilon > 0$), on a*

$$\begin{aligned} f(t_0 \pm H) &= \sum_{k=0}^{m} \frac{f^{(k)}(t_0)}{k!}\, (\pm H)^k + \frac{1}{m!} \int_{t_0}^{t_0 \pm H} (t_0 \pm H - s)^m f^{(m+1)}(s)\, ds \\ &= \sum_{k=0}^{m} \frac{f^{(k)}(t_0)}{k!}\, (\pm H)^k + \frac{(\pm H)^{m+1}}{m!} \int_0^1 (1-\tau)^m f^{(m+1)}(t_0 \pm \tau H)\, d\tau \end{aligned} \tag{3.4}$$

3.1.2. Différences divisées dans le cadre 1D. Nous allons ici voir comment réaliser une version décentrée de la formule de Taylor-Lagrange, les coefficients de Taylor $f^{(k)}(t_0)/k!$ se trouvant remplacés par des substituts adaptés aux mathématiques discrètes (et en particulier à l'interpolation de Lagrange), les *différences divisées*.

On considère un segment $[a,b] = [t_0 - H, t_0 + H]$ de $\mathbb{R}$ et ce que nous appellerons un « *maillage* » de ce segment, c'est-à-dire une suite

$$x_0 = a < x_1 < \cdots < x_N = b$$

induisant une subdivision du segment $[a,b]$. On ne fait ici aucune hypothèse sur les x_n, hormis le fait qu'ils soient distincts et indexés de manière croissante de manière à balayer le champ d'étude $[t_0 - H, t_0 + H] = [a,b]$. Ces points x_n, $n = 0, ..., N$ (qui sont appelés les *nœuds* du maillage) ne sont en particulier pas supposés ici régulièrement espacés..

Si $y_0, ..., y_N$ sont $N+1$ nombres complexes, on introduit le polynôme d'interpolation de Lagrange

$$\text{Lagrange}[x_0, ..., x_N \,;\, y_0, ..., y_N](X)$$

interpolant la valeur y_n au point x_n, $n = 0, ..., N$. Ce polynôme de degré exactement N s'exprime (voir par exemple [**Y1**], Section 3.5.2) sous la forme

$$\begin{aligned} &\text{Lagrange}[x_0, ..., x_N \,;\, y_0, ..., y_N](X) = y[x_0] + y[x_0, x_1]\,(X - x_0) \\ &+ y[x_0, x_1, x_2]\,(X - x_0)(X - x_1) + \cdots \\ &\cdots + y[x_0, x_1, ..., x_{N-1}] \prod_{j=0}^{N-2} (X - x_j) + y[x_0, x_1, ..., x_N] \prod_{j=0}^{N-1} (X - x_j), \end{aligned} \tag{3.5}$$

où les nombres complexes $y[x_0, ..., x_n]$, $n = 0, ..., N$, sont appelés *différences divisées* des y_n par les x_n (dans l'ordre imposé $n = 0, ..., N$) et sont à extraire d'un tableau de nombres organisé suivant les règles

$$\begin{aligned} &y[x_n] = x_n \\ &y[x_0, ..., x_n] = \frac{y[x_0, ..., x_{n-1}] - y[x_1, ..., x_n]}{x_0 - x_n}\ , \qquad n = 0, ..., N. \end{aligned} \tag{3.6}$$

Le calcul des différences divisées s'organise suivant un algorithme triangulaire identique à celui qui a été construire le tableau de Richardson (voir (2.18)). La colonne d'indice k ($k = 0, ..., N$) de ce tableau représente la suite des nombres $u_n^{(k)} := y[x_n, x_{n+1}, ..., x_{n+k}]$, $n = 0, ..., N - k$. Cette colonne d'indice N est donc réduite à un élément ($y[x_0, ..., x_N]$) tandis que la colonne d'indice $k = 0$ est la colonne des entrées $y_n = y[x_n]$, $n = 0, ..., N$. Il est commode de placer (pour mémoire) en position $k = -1$ la colonne des nœuds $x_0, ..., x_N$, colonne qui sera rappelée à chaque étape de la progression vers la droite dans la construction du tableau. On passe en effet de la colonne d'indice $k \geq 0$ à la colonne suivante (d'indice $k+1$) par la règle

$$u_n^{(k+1)} = \frac{u_n^{(k)} - u_{n+1}^{(k)}}{x_n - x_{n+k+1}}\ ,\ n = 0, ..., N - k - 1.$$

La présentation est donc la suivante :

$$
\begin{array}{lllll}
\underline{x_0} & y_0 = y[x_0] & & & \\
\underline{x_1} & y_1 = y[x_1] & y[x_0,x_1] & & \\
\underline{x_2} & y_2 = y[x_2] & y[x_1,x_2] & y[x_0,x_1,x_2] & \\
\vdots & \vdots & \vdots & \vdots\searrow & \\
\vdots & \vdots & \vdots & \vdots & y[x_0,...,x_N] \\
\vdots & \vdots & \vdots & \vdots\nearrow & \\
\underline{x_{N-2}} & y_{N-2} = y[x_{N-2}] & y[x_{N-2},x_{N-1}] & y[x_{N-2},x_{N-1},x_N] & \\
\underline{x_{N-1}} & y_{N-1} = y[x_{N-1}] & y[x_{N-1},x_N] & & \\
\underline{x_N} & y_N = y[x_N] & & &
\end{array}
$$

On remarque que le calcul de la différence divisée finale $y[x_0,...,x_N]$ est indépendant de l'ordre dans lequel on organise les nœuds x_n, $n=0,...,N$, pourvu que les y_n, $n=0,...,N$ soient organisés suivant la même permutation de $\{0,...,N\}$.

Une version « décentrée » de la formule de Taylor-Lagrange (Proposition 3.2), souvent utile en algorithmique numérique, s'énonce alors comme suit :

PROPOSITION 3.4 (version décentrée de la formule de Taylor-Lagrange). *Soit $t_0 \in \mathbb{R}$, $H>0$, et f une fonction de classe C^{m+1} sur l'intervalle $[t_0-H, t_0+H]$, à valeurs réelles. Soient $x_0,...,x_m$ $m+1$ points distincts du segment $[t_0-H,t_0+H]$.*

$$
\begin{aligned}
& \forall\, h \in]-H,H[,\ \exists, \xi_{x,h} \in]t_0-H, t_0+H[, \\
f(t_0+h) &= f[x_0] + \sum_{k=1}^{m} f[x_0,...,x_k] \prod_{j=0}^{k-1}(t_0+h-x_j) + \\
&\qquad + \frac{f^{(m+1)}(\xi_{x,h})}{(m+1)!}\prod_{j=0}^{m}(t_0+h-x_j) \\
&= \text{Lagrange}\,[x_0,...,x_m\,;\, f(x_0),...,f(x_m)]\,(t_0+h) + \\
&\qquad + \frac{f^{(m+1)}(\xi_{x,h})}{(m+1)!}\prod_{j=0}^{m}(t_0+h-x_j), \qquad (3.7)
\end{aligned}
$$

où les $f[x_0,...,x_k]$, $k=0,...,m$, dénotent les différences divisées $y_f[x_0,...,x_k]$, $k=0,...,m$, avec $y_{f,n}=f(x_n)$ pour $n=0,...,m$.

DÉMONSTRATION. La preuve de la Proposition 3.4 repose sur une application répétée du théorème de Rolle à la fonction

(3.8)

$$
\begin{aligned}
& t \mapsto \text{Lagrange}\,[x_0,...,x_m\,;\, f(x_0),...,f(x_m)]\,(t) - f(t) + \\
& + \Big(f(t_0+h) - \text{Lagrange}\,[x_0,...,x_m\,;\, f(x_0),...,f(x_m)]\,(t_0+h)\Big)\prod_{j=0}^{n}\frac{t-x_j}{t_0+h-x_j}.
\end{aligned}
$$

On suppose ici que t_0+h est distinct de tous les x_j (dans le cas contraire, la formule (3.7) est immédiate). Cette fonction admet $m+1$ zéros distincts dans $[t_0-H,t_0+H]$. Le théorème de Rolle appliqué m fois assure que que sa dérivée d'ordre $m+1$ s'annule en un point $\xi_{x,h}$ de $]t_0-H,t_0+H[$, ce qui donne le résultat voulu. □

REMARQUE 3.3. Il faut noter que l'on a toujours (quelque soit la fonction f définie sur $[t_0 - H, t_0 + H]$, qu'elle soit à valeurs dans $\mathbb{R}$ ou $\mathbb{C}$, même à valeurs vectorielles, qu'elle soit régulière ou non) l'identité algébrique

$$\begin{aligned} \forall h \in]-H, H[,\ f(t_0+h) &= f[x_0] + \sum_{k=1}^{m} f[x_0, ..., x_k] \prod_{j=0}^{k-1} (t_0 + h - x_j) \\ &\quad + f[x_0, ..., x_m, t_0 + h] \prod_{j=0}^{m} (t_0 + h - x_j), \end{aligned} \tag{3.9}$$

où $f[x_0, ..., x_m, t_0 + h]$ dénote la différence divisée $y_f[x_0, ..., x_n, t_0 + h]$ (il s'agit d'un vecteur de $\mathbb{R}^p$ si f est à valeurs dans $\mathbb{R}^p$) avec $y_{f,n} = f(x_n)$ pour $n = 0, ..., m$ et $y_{f,m+1} = f(t_0 + h)$.

3.2. Calcul numérique d'une dérivée 1D (approximation linéaire)

Étant donnée une fonction f de classe C^1 sur $[a, b]$, à valeurs réelles, nous allons expliquer comment évaluer la dérivée de f en un point donné ξ de $[a, b]$. L'application

$$f \in C^1([a, b]) \longmapsto f'(\xi)$$

est clairement une application linéaire de $C^1([a, b], \mathbb{R})$ dans $\mathbb{R}$. Supposer $h = b - a$ petit revient à supposer [3] que l'on ne dispose pour faire ce calcul approché que des valeurs de f aux points a et b, points que l'on peut considérer comme les points d'un maillage à deux nœuds de $[a, b]$: $x_0 = a < x_1 = b$. La valeur approchée de $f'(\xi)$ se doit donc d'être nécessairement de la forme

$$[f'(\xi)]_{\text{app}} = \lambda_\xi f(a) + \mu_\xi f(b),$$

où λ_ξ et μ_ξ sont deux scalaires donnés (indépendants de $f \in C^1([a, b], \mathbb{R})$) ; il faut en effet respecter au niveau du calcul approché le fait que l'évaluation de la dérivée en un point donné ξ dépende de manière linéaire de la fonction C^1 de départ.

Pour décider de ce calcul approché du nombre $f'(\xi)$, nous allons introduire un sous-espace $\mathcal{L}_2$ de dimension 2 (le nombre de nœuds du maillage considéré, ici en l'occurrence 2) de $C^1([a, b], \mathbb{R})$, assez riche pour modéliser raisonnablement sur l'intervalle d'étude $[a, b]$ les éléments du $\mathbb{R}$-espace vectoriel $C^1([a, b], \mathbb{R})$. Nous convenons d'adapter les coefficients λ_ξ et μ_ξ de manière à ce que le calcul approché devienne un calcul exact lorsque $f \in \mathcal{L}_2$, c'est-à-dire

$$\forall\, f \in \mathcal{L}_2 \subset C^1([a, b], \mathbb{R}),\ f'(\xi) = [f'(\xi)]_{\text{app}} = \lambda_\xi f(a) + \mu_\xi f(b).$$

Compte tenu de ce que $h = b - a$ est petit, il est raisonnable ici de considérer comme sous-espace $\mathcal{L}_2$ le sous espace des fonctions affines sur $[a, b]$. Ce choix particulier de $\mathcal{L}_2$ (exploitable uniquement si le maillage est à deux points) correspond à ce que l'on appelle le procédé d'*approximation linéaire*. Comme le sous-espace des fonctions affines est engendré par les fonctions $t \mapsto 1$ et $t \mapsto t$, les réels λ_ξ et μ_ξ sont donnés par les deux conditions

$$0 = \lambda_\xi + \mu_\xi \quad \& \quad 1 = \lambda_\xi a + \mu_\xi b.$$

3. Ce « pas » $h = b - a$ correspond par exemple au pas d'échantillonnage de l'information analogique que l'on étudie sous forme discrétisée en l'échantillonnant.

Quelque soit la valeur de ξ, on constate que la valeur approchée de $f'(\xi)$ est donnée par

$$[f'(\xi)]_{\text{app}} = \frac{f(b)-f(a)}{b-a}.$$

L'erreur commise entre $f'(\xi)$ et sa valeur approchée est donnée par

$$E_\xi(f) = f'(\xi) - [f'(\xi)]_{\text{app}} = f'(\xi) - \frac{f(b)-f(a)}{b-a}.$$

Cette erreur dépend, elle, du point ξ choisi (quand bien même la valeur approchée choisie pour $f'(\xi)$ n'en dépend pas). Suivant que l'on choisisse $\xi = (a+b)/2$ (ce qui correspond pour le calcul numérique approché de calcul de dérivée au *schéma numérique centré*), ou $\xi \neq (a+b)/2$ (on parle alors pour ce calcul numérique approché de calcul de dérivée de *schéma numérique décentré*) , l'estimation d'erreur sera en

$$|E_{(a+b)/2}(f)| \leq \frac{h^2}{24} \sup_{[a,b]} |f'''| \tag{3.10}$$

(pourvu que f soit au moins C^3 sur $[a,b]$) dans le cas centré $\xi = (a+b)/2$, tandis que l'on ne pourra dans le second cas espérer mieux qu'une estimation en

$$|E_\xi(f)| \leq \frac{h}{2} \sup_{[a,b]} |f''| \tag{3.11}$$

(pourvu que f soit au moins C^2 sur $[a,b]$) lorsque $\xi \neq (a+b)/2$. La raison principale en est que dans le cas centré, la formule

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

reste « miraculeusement » vraie pour les fonctions polynomiales de degré 2 car

$$2\xi = a+b = \frac{b^2-a^2}{b-a}$$

(alors qu'on ne l'avait pas *a priori* exigé !) alors que ceci est faux pour le schéma numérique décentré car dans ce cas

$$2\xi \neq a+b = \frac{b^2-a^2}{b-a}.$$

Pour obtenir la majoration d'erreur (3.10) dans le cas centré, on utilise pour représenter f sur $[a,b]$ (lorsque f est au moins C^3) la formule de Taylor avec reste intégral (3.4) à l'ordre $m=2$. L'erreur $E_{(a+b)/2}(f)$ est ainsi la même que celle que l'on commet en remplacant f par

$$t \mapsto f(t) - f(a) - tf'(a) - \frac{t^2}{2}f''(a) = \frac{1}{2}\int_a^t (t-s)^2 f'''(s)\,ds.$$

On a donc dans le cas centré :

$$\begin{aligned} E_{(a+b)/2}(f) &= \frac{1}{2}E_{(a+b)/2}\Big[t \mapsto \int_a^t (t-s)^2 f'''(s)\,ds\Big] \\ &= -\frac{1}{2h}\Big(\int_a^{(a+b)/2} (a-s)^2\, f'''(s)\,ds + \int_{(a+b)/2}^b (b-s)^2\, f'''(s)\,ds\Big), \end{aligned}$$

d'où la majoration d'erreur (3.10) dans ce cas.

Dans le cas décentré, on doit se contenter d'utiliser la formule de Taylor avec reste intégral (3.4) seulement à l'ordre 1. L'erreur $E_\xi(f)$ est ainsi la même que celle que l'on obtient en remplacant f par

$$t \mapsto f(t) - f(a) - tf'(a) = \int_a^t (t-s)f''(s)\,ds.$$

On a donc dans ce schéma décentré :

$$\begin{aligned} E_\xi(f) &= E_\xi\Big[t \mapsto \int_a^t (t-s)f''(s)\,ds\Big] \\ &= \frac{1}{h}\Big(\int_a^\xi (s-a)\,f'''(s)\,ds + \int_\xi^b (s-b)\,f''(s)\,ds\Big), \end{aligned}$$

d'où l'estimation d'erreur (3.11) dans ce cas.

En conclusion, le choix du schéma numérique centré

$$f'((a+b)/2) \simeq \frac{f(b)-f(a)}{b-a}$$

est à privilégier par rapport au choix du schéma numérique

$$[f'(\xi)]_{\text{app}} = \frac{f(b)-f(a)}{b-a}\,, \quad \xi \neq (a+b)/2$$

lorsqu'il s'agit de calculer la dérivée en un point à partir d'un maillage à deux nœuds par approximation linéaire (l'espace $\mathcal{L}_2$ étant l'espace des fonctions polynomiales de degré au plus 1, *i.e* l'espace des fonctions affines). Dans le premier cas en effet, l'erreur commise (pourvu que f soit assez régulière) est en $O(h^2)$, tandis que dans le second elle est en $0(h)$.

3.3. Dérivation versus intégration : la formule d'Euler-MacLaurin

L'opération inverse du calcul numérique de dérivées est la primitivisation discrète. Comme on l'a vu dans la section précédente, le calcul numérique de la dérivée d'une fonction en un point ξ d'un segment $[a, a+h]$ se résume à

$$[f'(\xi)]_{\text{app}} = \frac{f(a+h)-f(a)}{h}.$$

Le schéma centré ($\xi = (a+b)/2$) étant, on l'a vu, à privilégier par rapport au schéma décentré. L'opération discrète de primitivisation (toujours par approximation linéaire, à partir d'un schéma à deux nœuds) se ramène au calcul d'intégrale par la *formule des trapèzes*

(3.12)

$$\Big[\int_a^{a+h} f(t)\,dt\Big]_{\text{app}} = \int_a^{a+h} \Big(f(a)+(t-a)\frac{f(a+h)-f(a)}{h}\Big)\,dt = \frac{h}{2}\,(f(a)+f(a+h)).$$

En écrivant la formule de Taylor avec reste intégral à l'ordre 1 :

$$f(t) = f(a) + tf'(a) + \int_a^t (t-s)f''(s)\,ds\,,\ t \in [a, a+h],$$

on voit que l'erreur $E_{\rm trap}(f)$ commise dans le calcul approché (3.12) vaut

$$\begin{aligned} E_{\rm trap}(f) &= E_{\rm trap}\Big[t \mapsto \int_a^t (t-s) f''(s)\, ds\Big] \\ &= \frac{1}{2}\int_a^{a+h} (a+h-t)^2 f''(t)\, dt - \frac{h}{2}\int_a^{a+h} (a+h-s)\, f''(s)\, ds \\ &= \frac{1}{2}\int_a^{a+h} \Big((a+h-t)^2 - h(a+h-t)\Big)\, f''(t)\, dt \\ &= -\frac{1}{2}\int_a^{a+h} (t-a)(a+h-t)\, f''(t)\, dt\,, \end{aligned}$$

d'où l'estimation

$$|E_{\rm trap}(f)| \leq \sup_{[a,a+h]} |f''| \times \int_a^{a+h} (t-a)(a+h-t)\, dt = \frac{h^3}{12} \sup_{[a,a+h]} |f''|. \tag{3.13}$$

L'erreur dans la primitivisation discrète *via* l'approximation linéaire sur $[a, a+h]$ (formule des trapèzes) s'avère donc en $O(|h|^3)$.

Ce processus de primitivisation discrète par approximation linéaire est intimement lié au mécanisme de dérivation par une formule combinatoire très importante, la *formule d'Euler-Maclaurin*. Pour introduire cette formule (avatar de la formule d'intégration par parties), il convient d'introduire la suite des *polynômes de Bernoulli*.

3.3.1. Nombres et polynômes de Bernoulli.

Définition 3.4 (nombres et polynômes de Bernoulli). Les nombres de Bernoulli b_n, $n = 0, 1, 2, ...$, sont déduits du développement en série entière au voisinage de l'origine de la fonction

$$z \mapsto \frac{z}{e^z - 1} = \frac{1}{1 + \frac{z}{2} + \frac{z^2}{6} + \frac{z^3}{24} + \cdots} = 1 - \frac{z}{2} + \frac{z^2}{6} + \cdots = \sum_{k=0}^{\infty} \frac{b_k}{k!}\, z^k. \tag{3.14}$$

Ces nombres se calculent inductivement *via* l'algorithme de division de polynômes suivant les puissances croissantes ($b_0 = 1$, $b_1 = -1/2$, $b_2 = 1/6$, $b_3 = 0$, $b_4 = -1/30$, $b_5 = 0$, $b_6 = 1/42$, $b_7 = 0$,...). On a $b_{2k+1} = 0$ pour tout $k \geq 1$. Les polynômes de Bernoulli B_n, $n = 0, 1, ...$ sont définis à partir des nombres de Bernoulli b_n, $n = 0, 1, ...$, par les formules combinatoires

$$B_n(X) = \sum_{k=0}^{n} \binom{n}{k} b_k X^{n-k}. \tag{3.15}$$

On a immédiatement $B_0(X) \equiv 1$, $B_1(X) = X - 1/2$. D'autre part, B_n est un polynôme unitaire de degré n tel que $B_n(0) = b_n$. D'autre part, comme

$$z = \sum_{k=1}^{\infty} \frac{z^k}{k!} \times \sum_{k=0}^{\infty} \frac{b_k}{k!}\, z^k$$

au voisinage de $z = 0$ (du fait de (3.14)), on en déduit (en effectuant le produit de Cauchy) les relations

$$\sum_{k=0}^{n-1} \binom{n}{k} b_k = 0\,,\ n = 1, 2, ...$$

Ainsi, on voit que $B_1(1) = 1/2$ et que $B_n(1) = B_n(0) = b_n$ lorsque $n > 1$. Une autre relation très importante liant la suite des polynômes de Bernoulli est le fait que

$$\frac{d}{dX}B_n = nB_{n-1}\,,\ n = 1, 2, ... \tag{3.16}$$

Il résulte de cette formule inductive que si f est une fonction de classe C^1 sur un segment $[k, k+1]$ avec $k \in \mathbb{N}^*$ et $m \geq 1$,

$$\begin{aligned}
&\int_k^{k+1} f(t)B_m(t-k)\,dt = \\
&= \frac{1}{m+1}\Big[f(t)B_{m+1}(t-k)\Big]_k^{k+1} - \frac{1}{m+1}\int_k^{k+1} f'(t)B_{m+1}(t)\,dt \\
&= \frac{b_{m+1}}{m+1}(f(k+1)-f(k)) - \frac{1}{m+1}\int_k^{k+1} f'(t)\,B_{m+1}(t-k)\,dt. \qquad (3.17)
\end{aligned}$$

3.3.2. La formule d'Euler-MacLaurin. La formule d'intégration par parties (3.17) (inhérente aux propriétés des polynômes de Bernoulli) permet d'établir par récurrence la première forme de la formule d'Euler-Maclaurin, dite *forme discrète* (permettant ultérieurement de relier les calculs de sommes de séries à des calculs de dérivées).

PROPOSITION 3.5 (formule d'Euler-Maclaurin, forme discrète). *Soit f une fonction de classe C^m sur $[1, N]$ ($m \in \mathbb{N}^*, N \in \mathbb{N}^*$), à valeurs réelles ou complexes. Alors*

$$\begin{aligned}
\sum_{k=1}^{N-1} f(k) = \int_1^n f(t)\,dt + \sum_{l=1}^{m}\frac{b_l}{l!}\Big(f^{(l-1)}(N) - f^{(l-1)}(1)\Big) + \\
+ \frac{(-1)^{m+1}}{m!}\int_1^N B_m(t)\,f^{(m)}(t - E[t])\,dt.
\end{aligned} \tag{3.18}$$

DÉMONSTRATION. Cette formule se démontre par récurrence sur m. Si $m = 1$, il résulte d'une intégration par parties que pour tout entier k entre 1 et $N-1$,

$$\int_k^{k+1} B_1\Big(t-k-\frac{1}{2}\Big)\,f'(t)\,dt = \frac{f(k)+f(k+1)}{2} - \int_k^{k+1} f(t)\,dt.$$

En sommant de $k = 1$ à $k = N-1$, il vient donc

$$\begin{aligned}
\int_1^n B_1(t - E[t])\,f'(t)\,dt &= \frac{1}{2}\sum_{k=1}^{N-1}(f(k)+f(k+1)) - \int_1^N f(t)\,dt \\
&= \sum_{k=1}^{N-1} f(k) + \frac{f(N)-f(1)}{2} - \int_1^N f(t)\,dt,
\end{aligned}$$

ce qui correspond à la formule d'Euler-MacLaurin discrète (3.18) dans le cas $m = 1$. Lorsque $m \geq 1$, on admet la formule (3.5) et on transforme

$$\frac{1}{m!}\int_1^N B_m(t - E[t])\,f^{(m)}(t)\,dt = \frac{1}{m!}\sum_{k=1}^{N-1}\int_k^{k+1} B_m(t-k)f^{(m)}(t)\,dt$$

suivant la formule d'intégration par parties (3.17) appliquée à chaque intégrale du membre de droite (avec f remplacée par $f^{(m)}$). Après addition on trouve

$$\begin{aligned}&\frac{1}{m!}\int_1^N B_m(t-E[t])\,f^{(m)}(t)\,dt = \\ = \;& \frac{b_{m+1}}{(m+1)!}\Big(f^{(m)}(N)-f^{(m)}(1)\Big) - \frac{1}{(m+1)!}\int_1^N B_{m+1}(t-E[t]),f^{(m+1)}(t)\,dt.\end{aligned}$$

En reportant ceci dans (3.5), on obtient une nouvelle formule, en fait la formule (3.5) encore, mais écrite cette fois au cran $m+1$ à la place de m. La formule d'Euler-MacLaurin (3.5) est ainsi prouvée par récurrence. □

La formule d'Euler MacLaurin sous forme discrète (3.5) se transforme par changement de variables en une forme (que nous qualifierons de « continue »), plus adapté non plus comme (3.5) au calcul numérique de sommes de séries, mais au calcul approché d'intégrales de fonctions continues par discrétisation. Voici cette version :

PROPOSITION 3.6 (formule d'Euler-Maclaurin, forme « continue »). *Soit f une fonction de classe C^{2m} sur un intervalle $[a,b]$ de $\mathbb{R}$ et $h := (b-a)/N$ pour $N \geq 1$. On a*

$$\begin{aligned}\int_a^b f(t)\,dt = h\Big(\frac{f(a)}{2} + \sum_{k=1}^{N-1} f(a+kh) + \frac{f(b)}{2}\Big) - \\ - \sum_{l=1}^{m} \frac{b_{2l}}{(2l)!}\Big(f^{(2l-1)}(b) - f^{(2l-1)}(a)\Big)\,h^{2l} + \\ + \frac{h^{2m}}{(2m)!}\int_a^b B_{2m}\Big(\frac{t-a}{h} + 1 - E\Big[\frac{t-a}{h}+1\Big]\Big)\,f^{(2m)}(t)\,dt.\end{aligned} \tag{3.19}$$

DÉMONSTRATION. On utilise le fait que les nombres de Bernoulli b_l sont nuls lorsque l est impair et l'on applique la formule (3.5) à l'ordre $2m$ (d'où la disparition du signe $-(-1)^{2m+1} = 1$ dans le dernier terme d'erreur. □

REMARQUE 3.5. Le premier terme au membre de droite de (3.19) représente la valeur approchée de l'intégrale, calculée sur chaque segment de la subdivision de $[a,b]$ par les $a+kh$, $k = 0, ..., N-1$, par la méthode des trapèzes. C'est sur cette formule que s'articulera ultérieurement le calcul numérique d'intégrales par la méthode de Romberg.

3.4. Calcul différentiel en plusieurs variables

3.4.1. Les formules de Taylor (version multi-D). Les trois formules de Taylor (Taylor-Young, Taylor-Lagrange, Taylor avec reste intégral) s'étendent naturellement au cadre multi-variables. Comme dans le contexte 1D, la formule fournissant le moins d'informations par rapport à l'algorithmique numérique est la formule de Taylor-Young, cette fois dans sa version multi-D.

PROPOSITION 3.7 (formule de Taylor-Young, version multi-D). *Si f est une fonction à valeurs réelles, complexes, ou vectorielles, définie au voisinage d'un point*

x_0 de $\mathbb{R}^n$, différentiable à l'ordre m en x_0, on a, pour h voisin de 0 dans $\mathbb{R}^n$,

$$f(x_0+h) = \sum_{\substack{\underline{l}\in\mathbb{N}^n \\ |\underline{l}|\leq m}} \frac{D^{\underline{l}}[f](x_0)}{l_1!\cdots l_n!}\, h_1^{l_1}\cdots h_n^{l_n} + o(\|h\|^m), \tag{3.20}$$

où $D^{\underline{l}}$ désigne l'opérateur différentiel

$$D^{\underline{l}} := \frac{\partial^{l_1}}{\partial x_1^{l_1}} \circ \cdots \circ \frac{\partial^{l_n}}{\partial x_n^{l_n}}$$

et $|\underline{l}| = l_1 + \cdots + l_n$ la longueur de ce multi-indice.

La seconde formule (formule de Taylor-Lagrange, version multi-D) fournit une information plus précise. Elle ne vaut cependant que pour les fonctions à valeurs réelles (sinon, il faut se contenter d'une inégalité).

Proposition 3.8 (formule de Taylor-Lagrange, version multi-D). *Si f est une fonction à valeurs réelles de classe C^m dans une boule fermée $\overline{B(x_0,R)}$, admettant une différentielle à l'ordre $m+1$ en tout point de $B(x_0,R)$, on a, pour tout $h\in\mathbb{R}^n$ tel que $\|h\|<R$,*

$$f(x_0+h) = \sum_{\substack{\underline{l}\in\mathbb{N}^n \\ |\underline{l}|\leq m}} \frac{D^{\underline{l}}[f](x_0)}{l_1!\cdots l_n!}\, h_1^{l_1}\cdots h_n^{l_n} + \frac{1}{(m+1)!}\mathbf{D}^{m+1}[f](\xi)\ \overset{(m+1)\ fois}{(h,...,h)}\,, \tag{3.21}$$

où $\mathbf{D}^{m+1}[f](\xi)$ désigne la différentielle d'ordre $m+1$ de f au point ξ (il s'agit, on le rappelle, voir par exemple [**MathL2**]*, chapitre 14, d'une forme $(m+1)$- linéaire symétrique sur $\mathbb{R}^{m+1}$). Lorsque f est à valeurs vectorielles, il faut se contenter de l'inégalité*

$$\Big\| f(x_0+h) - \sum_{\substack{\underline{l}\in\mathbb{N}^n \\ |\underline{l}|\leq m}} \frac{D^{\underline{l}}[f](x_0)}{l_1!\cdots l_n!}\, h_1^{l_1}\cdots h_n^{l_n} \Big\| \leq \frac{\|h\|^{m+1}}{(m+1)!} \sup_{\xi\in]x_0,x_0+h[} \|\mathbf{D}^{m+1}[f](\xi)\|. \tag{3.22}$$

La plus importante des trois formules (du point de vue qui nous concerne, à savoir l'algorithmique numérique) demeure la plus explicite des trois, à savoir la formule de Taylor avec reste intégral.

Proposition 3.9 (formule de Taylor avec reste intégral, version multi-D). *Soit f une fonction de classe C^m dans la boule fermée $B(x_0,R)$, à valeurs réelles, complexes, ou dans $\mathbb{R}^p$. Pour tout H dans $\mathbb{R}^n$ tel que $\|H\|\leq R$, on a*

$$\begin{aligned} f(x_0+H) = &\sum_{\substack{\underline{l}\in\mathbb{N}^n \\ |\underline{l}|\leq m}} \frac{D^{\underline{l}}[f](x_0)}{l_1!\cdots l_n!}\, H_1^{l_1}\cdots H_n^{l_n} + \\ &+ \frac{1}{m!}\int_0^1 (1-\tau)^m\, \mathbf{D}^{m+1}[f](x_0+\tau H)\ \overset{(m+1)\ fois}{(H,...,H)}\ d\tau. \end{aligned} \tag{3.23}$$

Une version décentrée de la formule de Taylor-Lagrange peut être envisagée dans un pavé produit d'intervalles $\overline{P} = [x_{01}-H_1, x_{01}+H_1]\times\cdots\times[x_{0n}-H_n, x_{0n}+H_n]$, étant donnée, pour chaque $j=1,...,n$, une suite de points distincts $\xi_j^{(0)},...,\xi_j^{(N_j)}$ du segment $[x_{0j}-H_j, x_{0j}+H_j]$. On effectue pour cela en n temps successifs un calcul de polynôme d'interpolation de Lagrange multi-D (exprimé à partir des différences

divisées successives, cette fois en plusieurs variables [4]). On note ce polynôme d'interpolation de Lagrange multi-D

$$\text{Lagrange}\left[\begin{pmatrix}\xi_1^{(0)},...,\xi_1^{(N_1)}\\ \vdots \\ \xi_n^{(0)},...,\xi_n^{(N_n)}\end{pmatrix}\ ;\ f(\xi_1^{(\bullet)},...,\xi_n^{(\bullet)})\right](X_1,X_2,...,X_n). \tag{3.25}$$

La formule de Taylor-Lagrange décentrée

$$f(x)\simeq \text{Lagrange}\left[\begin{pmatrix}\xi_1^{(0)},...,\xi_1^{(N_1)}\\ \vdots \\ \xi_n^{(0)},...,\xi_n^{(N_n)}\end{pmatrix}\ ;\ f(\xi_1^{(\bullet)},...,\xi_n^{(\bullet)})\right](x),\ x\in P, \tag{3.26}$$

pose beaucoup de problèmes quant au contrôle d'erreur (il en est de même pour la formule de Lagrange-Kronecker-Jacobi $f(x)\simeq Q_f[P_1,...,P_n](x)$, où $Q_f[P_1,...,P_n]$ est le polynôme d'interpolation de Lagrange-Jacobi-Kronecker donné par (3.24)). Le contrôle d'erreur dont on dispose est en effet certes hérité de celui donné par la formule de Taylor-Lagrange décentrée 1D (Proposition 3.4), mais il faut l'itérer en suivant le mécanisme conduisant (en raisonnant variable après variable) à l'obtention de la formule approchée (3.26) ! Nous profitons de mentionner cette difficulté pour souligner que les *méthodes d'éléments finis* (dont nous parlerons ultérieurement) s'avèrent numériquement plus efficaces que la méthode d'interpolation basée sur l'extension au cadre multi-D du procédé d'interpolation de Lagrange. Les fonctions *spline* (*i.e* affines ou plus généralement polynomiales par morceaux en les diverses variables $x_1,...,x_n$) que nous introduirons au chapitre 4 (section 4.1) seront alors privilégiées aux fonctions polynômes ; elles autoriseront en effet beaucoup plus de souplesse que n'en autorise l'extrême rigidité du cadre algébrique des fonctions polynômes.

3.4.2. Méthode de Newton. Soit f une fonction à valeurs réelles de classe C^1 sur un segment de l'axe réel $[t_0-H,t_0+H]$, plus généralement une fonction F à valeurs dans $\mathbb{R}^n$, de classe C^1 dans une boule fermée $\overline{B(x_0,R)}$ (par exemple $F=(\mathrm{Re}\, f,\mathrm{Im}\,(f))$, où f est une fonction holomorphe de $\overline{D(z_0,R)}$ dans $\mathbb{C}$). Il est important du point de vue de l'algorithmique numérique de savoir calculer de

4. Il suffit d'enchainer les mécanismes $1D$. On note toutefois l'inconvénient dans cette démarche d'interpolation de Lagrange multi-D de ne pouvoir travailler qu'avec un réseau cartésien de points. De fait, travailler avec une grille de points distincts $\xi_1,...,\xi_N$ du pavé $\overline{P}$ définis comme les zéros communs dans $\overline{P}$ de n polynômes $P_1,...,P_n$ de n variables ne présentant dans $\overline{P}$ que des zéros simples (*i.e* où le déterminant jacobien $\mathrm{jac}\,(P_1,...,P_n)$ de $(P_1,...,P_n)$ ne s'annule pas), pourrait aussi être envisagé lorsque f est une fonction analytique des variables x. Cela conduit (au lieu du polynôme de Lagrange multi-D mentionné en (3.25)) à la construction du *polynôme de Lagrange-Jacobi-Kronecker* :

$$Q_f[P_1,...,P_n](X)=\sum_{j=1}^{N}\frac{f(\xi_j)}{\mathrm{jac}(P_1,...,P_n)(\xi_j)}\ \det[Q_{jk}(X,\xi_j)]_{1\le j,k\le n}\,, \tag{3.24}$$

où les Q_{jk} (par exemple calculables suivant le principe de calcul des différences divisées) sont construits de manière à satisfaire les identités polynomiales

$$P_k(X)-P_k(Y)=\sum_{j=1}^{n}Q_{kj}(X,Y)\,(X_j-Y_j),\ k=1,...,n.$$

manière approchée les zéros éventuels de la fonction f (ou les points où $F(x) = (0, ..., 0)$). Ces points sont aussi, notons le, les points fixes de l'application

$$(x_1, ..., x_n) \mapsto (F_1(x) + x_1, ..., F_n(x) + x_n)$$

et on verra ultérieurement comment dans certains cas une méthode itérative basée sur le théorème du point fixe. La *méthode de Newton* proposée ici (et qui, plus tard, inspirera la méthode du gradient conjugué) est, elle, fondée sur la « linéarisation » de la fonction F, consistant à remplacer la fonction F dans $B(x_0, R)$ par son approximation affine

$$x \mapsto F(x_0) + \mathbf{D}[F](x_0)(x - x_0).$$

Lorsque $\mathbf{D}[F](x_0)$ est inversible, cette linéarisée s'annule en un seul point, à savoir le point donné par

$$x = x_0 - (\mathbf{D}[F](x_0))^{-1} (F(x_0))$$

(x et $F(x)$ sont ici pensés comme des vecteurs colonne). C'est sur cette idée simple que s'appuie le principe de l'algorithme itératif de Newton. Si l'on peut mettre en œuvre un mécanisme itératif initié en un point $X_0 = x_{\text{init}}$ de $B(x_0, R)$ et tel que la suite récurrence obéissant à la règle

$$X_{k+1} = X_k - (\mathbf{D}[F](X_k))^{-1} F(X_k)$$

puisse être définie (ce qui suppose qu'à chaque cran de l'algorithme, le point X_k trouvé reste dans $B(x_0, R)$ et de plus soit tel que $\mathbf{D}[F](X_k)$ soit inversible), on voit en effet que, si la suite $(X_k)_{k\geq 0}$, sa limite est un point de $\overline{B(x_0, R)}$ où F s'annule.

Pour cela, nous disposons de la proposition « garde-fou » suivante.

PROPOSITION 3.10. *Soit F une fonction de classe C^2 dans $\overline{B(\xi, r)} \subset \mathbb{R}^n$, à valeurs dans $\mathbb{R}^n$, telle que $F(\xi) = 0$ et que $\mathbf{D}[F](x)$ soit inversible (i.e de déterminant jacobien non nul) pour tout $x \in \overline{B(\xi, r)}$. Soit γ la quantité définie par*

$$\gamma := \sup_{x, x' \in \overline{B(\xi, r)}} \|\mathbf{D}^2[F](x)\| \, \|(\mathbf{D}[F](x'))^{-1}\| \tag{3.27}$$

(la norme d'application linéaire de $\mathbb{R}^n$ dans $\mathbb{R}^n$ choisie ici est toujours la norme $\| \ \|_2$, en accordance avec la norme euclidienne usuelle $\| \ \|_2$ dans $\mathbb{R}^n$). Si $\gamma r < 2$ et si $x \in \overline{B(\xi, x)}$, on a

$$N_F(x) := x - (\mathbf{D}[F](x))^{-1} (F(x)) \in \overline{B(\xi, r)}$$

avec de plus

$$\|N_F(x) - \xi\| \leq \frac{\gamma}{2} \|x - \xi\|^2. \tag{3.28}$$

DÉMONSTRATION. On a, de part la formule de Taylor avec reste intégral écrite à l'ordre 1,

$$\begin{aligned} N_F(x) - \xi &= (x - \xi) - (\mathbf{D}[F](x))^{-1}(F(x) - F(\xi)) \\ &= -(\mathbf{D}[F](x))^{-1} \Big(F(x) - F(\xi) - \mathbf{D}[F](x) \, (x - \xi) \Big) \\ &= -(\mathbf{D}[F](x))^{-1} \left[\int_0^1 (1 - \tau) \, \mathbf{D}^2[F](\xi + \tau(x - \xi))(x - \xi, x - \xi) \, d\tau \right]. \end{aligned}$$

En prenant les normes, on en déduit (par inégalité triangulaire), l'estimation

$$\begin{aligned}\|N_F(x)-x\| &\leq \int_0^1 (1-\tau)\,\|(\mathbf{D}[F](x))^{-1}\|\,\|\mathbf{D}^2[F](\xi+\tau(x-\xi))\|\,\|x-\xi\|^2\,d\tau\\ &\leq \gamma\|x-\xi\|^2\int_0^1(1-\tau)\,d\tau\\ &\leq \frac{\gamma}{2}\|x-\xi\|^2 \leq \frac{\gamma r}{2}\|x-\xi\|.\end{aligned}$$

La conclusion de la proposition en résulte. □

Cette méthode soutend la validité de l'algorithme de Newton : si $X_0 = x_{\text{init}}$ est un point de $\overline{B(\xi,r)}$ et si $\gamma r < 2$ (γ étant définie en (3.27)), la suite récurrente générée à partir de X_0 par :

$$X_{k+1} = X_k - (\mathbf{D}[F](X_k))^{-1}\,(F(X_k))$$

est bien définie, converge vers le point ξ, ce avec le contrôle d'erreur :

$$\|X_k-\xi\| \leq \Big(\frac{\gamma r}{2}\Big)^{2^k-1}\|x_{\text{init}}-X_0\|. \tag{3.29}$$

REMARQUE 3.6. Lorsque $n=1$ et que la fonction f est telle que ni f', ni f'' ne s'annulent sur $[t_0-H,t_0+H]$, mais que $f(t_0-H)\times f(t_0+H)<0$, la fonction f s'annule en un unique point ξ de $[t_0-H,t_0+H]$ de par le théorème des valeurs intermédiaires. On constate alors sur une étude graphique (voir [**Y1**], Section 2.2) que l'algorithme de Newton, initié soit en $x_{\text{init}} = t_0-H$, soit en $x_{\text{init}} = t_0+H$ suivant les cas (il suffit dans chaque cas de faire le dessin, ce dépend du sens de monotonie et de la concavité ou convexité de f), génère une suite monotone de points de $[t_0-H,t_0+H]$ convergent en croissant ou en décroissant vers l'unique zéro ξ de f dans $[t_0-H,t_0+H]$. Il convient toutefois de prendre garde que, quand bien même f est strictement monotone et de classe C^2 dans $[t_0-H,t_0+H]$ avec $f(t_0-H)f(t_0+H)<0$, il est parfaitement possible que l'algorithme de Newton initié à l'une des extrémités t_0-H ou t_0+H ne converge pas : si par exemple $t_0-H=-1$, $t_0+H=1$, $f(t)=(5t-t^3)/4$ (il y a, notons le, changement de concavité sur l'intervalle car inflexion en $t=0$), cet algorithme ne fait que « rebondir » entre les valeurs $t=-1$ et $t=1$ sans bien sûr converger vers $t=0$!

REMARQUE 3.7. Si f est une fonction holomorphe au voisinage de $\overline{D(\xi,r)}$ s'annulant en ξ, telle que f' ne s'annule pas dans $\overline{D(\xi,r)}$, la constante γ définie en (3.27) vaut

$$\gamma = \sup_{\zeta,\zeta'\in\overline{D(\xi,r)}} \frac{|f''(\zeta)}{|f'(\zeta')|}.$$

Si $\gamma r<2$, la suite de nombres complexes initiée en un point z_0 de $\overline{D(\xi,r)}$ et générée par la règle inductive

$$z_{k+1} = z_k - \frac{f(z_k)}{f'(z_k)}$$

converge donc vers ξ avec

$$|z_k-\xi| \leq \Big(\frac{\gamma r}{2}\Big)^{2^k-1}|z_0-\xi|.$$

3.4.3. Application à l'optimisation (libre ou avec contraintes). Étant donnée une fonction f de classe C^2 dans un ouvert U de $\mathbb{R}^m$, on sait que les extréma locaux (et globaux) éventuels de la fonction f dans l'ouvert U sont à chercher parmi les points $x_0 \in U$ annulant le vecteur $\nabla f = (\partial f/\partial x_1, ..., \partial f/\partial x_n)$. Bien sûr, ces points ne sont peut-être pas tous des extréma locaux ou globaux ; parmi eux, par exemple, peuvent figurer des *points selle* (*cf.* l'exemple de $(0,0)$ pour $f(x,y) = x^2 - y^3$). Néanmoins, la recherche algorithmique de ces points ξ correspondant aux positions « éventuelles » des extréma locaux ou globaux de f peut être conduite suivant l'algorithme de Newton. Si $F = \nabla f$, la matrice de la différentielle $\mathbf{D}[\nabla f](x) : \mathbb{R}^n \to \mathbb{R}^n$ au point x est la *matrice Hessienne* de f

$$H_f(x) = \Big[\frac{\partial^2 f}{\partial x_j \partial x_k}\Big]_{1\leq j,k\leq n}$$

et l'algorithme de Newton se traduit alors par

$$\Big(x - [H_f(x)]^{-1}(\nabla[f](x))\Big) \longleftarrow x, \tag{3.30}$$

(x et $\nabla[f](x)$ étant pensés ici comme des vecteurs colonne), pourvu bien sûr que la matrice H_f soit inversible au point x.

Si l'on introduit des contraintes matérialisées par les $p \leq n$ équations

$$h_j(x_1, ..., x_n) = 0,\ j = 1, ..., p,$$

où $h_1, ..., h_p$ sont des fonctions de classe C^1 dans U, les points x_0 de U où f présente un extrémum local (ou global) sous les contraintes $h_j(x_1, ..., x_n) = 0$, $j = 1, ..., p$, sont à prendre parmi les points de U où l'on a à la fois

$$h_1(x_1, ..., x_n) = \cdots = h_p(x_1, ..., x_n) = 0 \tag{3.31}$$

(les contraintes sont satisfaites) et où le vecteur $\nabla f(x_0)$ appartient au sous-espace vectoriel engendré par les vecteurs $\nabla h_j(x_0)$, $j = 1, ..., p$ (que l'on suppose de dimension p en tous les points de U où les contraintes (3.31) sont remplies) [5]. La recherche de tels points x_0 (points candidats à localiser les points où f admet un éventuel extrémum local ou global sous les contraintes (3.31)) se ramène donc à la recherche des points $\Xi = (x, u_1^*, ..., u_p^*)$ dans $U \times \mathbb{R}^p$ solutions du système de $(n+p)$ équations à $(n+p)$ inconnues (x, u) :

$$\begin{aligned} h_1(x) &= 0 \\ &\vdots \\ h_p(x) &= 0 \\ \nabla f(x) + \sum_{j=1}^{p} u_j^* \nabla h_j(x) &= 0. \end{aligned} \tag{3.32}$$

La notation u^* matérialise le fait que ces variables $u_1^*, ..., u_p^*$ jouent le rôle de « variables duales ». Ceci équivaut à la recherche des points $\Xi = (x, u^*) \in \mathbb{R}^n \times \mathbb{R}^p$ solutions de

$$\nabla \mathbf{L}_{f;h}(\Xi) = 0,$$

5. Voir par exemple [**MathL2**], Chapitre 14, Section III.3.

où le *Lagrangien* $\mathbf{L}_{f;h}$ de f sous les contraintes $h_j = 0$, $j = 1, ..., p$, est donné par

$$\mathbf{L}_{f;h}(x, u^*) = f(x) + \sum_{j=1}^{p} u_j^* \, h_j(x).$$

On peut à nouveau utiliser pour approcher ces points Ξ (et donc les points x de U où la fonction f est candidate à présenter un extrémum local ou global sous les contraintes $h_j = 0$, $j = 1, ..., p$) la méthode de Newton (dans $\mathbb{R}^{n+p}$ cette fois) basée ici sur l'itération

$$\left(\begin{pmatrix} x \\ u^* \end{pmatrix} - [H^{-1}_{L_{f;h}}(x, u^*)](\nabla L_{f;h}(x, u^*)) \right) \longleftarrow \begin{pmatrix} x \\ u^* \end{pmatrix} \tag{3.33}$$

(les vecteurs x, u^* et $\nabla L_{f;h}(x, u*)$ sont ici pensés comme vecteurs colonne).

3.4.4. Méthode de descente et optimisation sans contraintes. L'attaque des problèmes d'optimisation sans ou avec contraintes *via* l'algorithme itératif de Newton suggère une attaque plus « géométrique » du problème, basée toujours sur l'idée d'approximation affine de la « fonction objectif » $f : \mathbb{R}^n \to \mathbb{R}$ à minimiser par son approximation affine

$$x \longmapsto f(x_0) + \mathbf{D}[f](x - x_0) = f(x_0) + \langle \nabla[f](x_0) \, , \, x - x_0 \rangle,$$

mais cette fois sur une interprétation différente de cette formule : si $\vec{u}$ est un vecteur unitaire arbitraire de $\mathbb{R}^n$, on compare les vitesses de décroissance (si décroissance il y a) des fonctions affines d'une variable

$$f_{\vec{u}} : t \in [0, \infty[\longmapsto f(x_0) + \mathbf{D}[f](x_0)(x_0 + t\vec{u} - x_0) = f(x_0) + t \, \langle \nabla f \, (x_0) \, , \, \vec{u} \rangle.$$

Pour qu'il y ait décroissance stricte d'une telle fonction $f_{\vec{u}}$ (et donc au niveau infinitésimal décroissance de $t \mapsto f(x_0 + t\vec{u})$), il faut et il suffit que $\langle \nabla f \, (x_0) \, , \, \vec{u} \rangle < 0$, le meilleur choix étant réalisé pour

$$\vec{u} = - \frac{\nabla f \, (x_0)}{\| \nabla f \, (x_0) \|}$$

(lorsque bien sûr $\nabla f \, (x_0)$ est non nul).

Algorithme 3 le gradient à pas optimal

```
x = x_0
si ||∇f(x)|| ≤ ε alors
    STOP
sinon
    p = min_{t∈[0,p_max]} f(x − t∇f(x))
    x = x − p ∇f(x)
    xi = gradient-pas-optimal (x, p_max, ε)
fin si
```

La méthode dite du *gradient à pas optimal* visant à approcher un point correspondant à un candidat « position de minimum » (pour la fonction objectif à minimiser f) est basée sur cette idée simple, traduite ici dans un langage de pseudo-code : on fixe un pas $p_{\max}$, de manière à contrôler les déplacements de x dans le déroulement

de algorithme [6], et un seuil $\epsilon > 0$ (conditionnant un test d'arrêt). C'est le prototype de ce que l'on qualifie de *méthode de descente*.

C'est le calcul du pas optimal p (ligne 5 de l'algorithme 2 présenté sous forme récursive ci-dessus), réactualisé à chaque étape, qui constitue le point algorithmiquement le plus délicat du processus. Pour trouver une approximation convenable ξ du point p où la fonction

$$t \in [0, p_{\max}] \longmapsto f(x - t\nabla f\,(x)) \tag{3.34}$$

atteint son minimum sur $[0, p_{\max}]$, une manière fréquente de procéder repose sur une idée algorithmique combinant méthode de Newton et méthode de dichotomie introduite dans les années 1950 par Frank Wolfe. Cette méthode peut être mise en route sur une fonction d'une variable φ régulière (disons C^1) sur un segment $[0, b]$ de $\mathbb{R}$, telle que $\varphi'(0) < 0$; on l'applique ici à la fonction (3.34) avec $b = p_{\max}$.

La méthode de F. Wolfe est initiée à partir du choix de deux seuils $0 < \gamma_1 < \gamma_2 < 1$ fixés une fois pour toutes. Pour comprendre le principe de cette méthode, il faut avoir en tête le schéma d'une fonction « cuvette » strictement convexe sur $[0, b]$, de dérivée strictement négative en 0, présentant son unique minimum sur $]0, b[$ (faites ici des dessins, en particulier dans les trois cas ou sous-cas envisagés dans la discussion ci-dessous !). On positionne la valeur de f au point médian $t_{\text{med}} = (t_{\text{init}} + t_{\text{ext}})/2$ de l'intervalle d'étude I (ici, pour commencer, $I = [0, b]$, ce qui donne $t_{\text{med}} = (0 + b)/2 = b/2$) par rapport au graphe de l'application affine

$$t \in I \longmapsto \varphi(0) + \gamma_1\, \varphi'(0)\, t.$$

Deux cas sont alors à distinguer :

(1) le point $(t_{\text{med}}, \varphi(t_{\text{med}}))$ est strictement au dessus de ce graphe, *i.e*

$$\varphi(t_{\text{med}}) > \varphi(0) + \gamma_1\, \varphi'(0)\, t_{\text{med}}.$$

On décide alors de poursuivre la méthode en prenant comme nouvel intervalle d'étude l'intervalle $[t_{\text{init}}, t_{\text{med}}] = [0, b/2]$ [7].

(2) le point $(t_{\text{med}}, \varphi(t_{\text{med}}))$ est au dessous de ce graphe, *i.e*

$$\varphi(0) + \gamma_1\, \varphi'(0)\, t_{\text{med}} \geq \varphi(t_{\text{med}}).$$

On distingue alors dans ce cas deux sous-cas de figure :

(a) on a $\varphi'(t_{\text{med}}) \geq \gamma_2\, \varphi'(0)$, auquel cas on décide de stopper l'algorithme de Wolfe et de choisir comme approximation du point p cherché [8] le point $\xi = t_{\text{med}} = b/2$;

(b) on a $\varphi'(t_{\text{med}}) < \gamma_2\, \varphi'(0)$, auquel cas on décide [9] de poursuivre l'algorithme en réinitialisant l'intervalle d'étude comme cette fois l'intervalle $[t_{\text{med}}, t_{\text{ext}}] = [b/2, b]$.

---

6. Il convient de veiller à prendre un pas $p_{\max}$ assez petit pour que seul le comportement de f dans un voisinage proche du point x influe sur le comportement des graphes des fonctions $f_{\vec{u}}$.

7. Ceci est logique, car, si la fonction φ était strictement convexe sur $[0, b]$, se trouver dans ce premier cas de figure impliquerait que l'unique minimum de φ sur $[0, b]$ est certainement atteint avant t_{med}.

8. Il s'agit, en fait, seulement d'une approximation d'un minimum local lorsque φ n'est pas convexe.

9. Ceci est, une fois encore, logique si l'on suppose φ strictement convexe sur $[0, b]$.

Si $f : \mathbb{R}^n \to \mathbb{R}$ est une fonction suffisamment régulière (disons pour simplifier de classe au moins C^2) sur laquelle on applique l'algorithme du gradient avec pas optimal intitié à partir d'un point x_0, on ne peut qu'espérer, en l'absence d'hypothèse forte de convexité, aboutir (dans les bons cas) en un point correspondant à un minimum local de f. Pour une suite de points $(x_k)_{k\geq 0}$ générée par cet algorithme de descente, on distingue trois types de configuration :

- la suite $(x_k)_{k\geq 0}$ est dite *stationnarisante* si l'on a
$$\lim_{k\to+\infty} \nabla f(x_k) = 0\,,$$
ce qui se produit dans le cas le plus fréquent (dans les bons cas, mais ce n'est pas la règle générale, on peut extraire de la suite $(x_k)_{k\geq 0}$ une sous-suite convergeant vers un point ξ où la fonction f présente un minimum local, ce point limite ξ dépendant bien sûr *a priori* du point initial x_0) ;
- la suite $(x_k)_{k\geq 0}$ est dite *minimisante* si
$$\lim_{k\to+\infty} f(x_k) = \inf_{\mathbb{R}^n} f\ ;$$
- la suite $(x_k)_{k\geq 0}$ est dite *converger vers une solution optimale* si elle converge vers un point ξ réalisant le minimum global de f sur $\mathbb{R}^n$.

Ce qui nous intéresse avant tout est de disposer d'un algorithme réalisant une suite convergeant vers une solution optimale. C'est le cas par exemple lorsque la fonction f *fortement convexe*, *i.e* telle que la matrice Hessienne de f
$$H_f = \Big[\frac{\partial^2 f}{\partial x_j \partial x_k}\Big]_{1\leq j,k\leq n}$$
ait (en tout point) toutes ses valeurs propres minorées par un réel $\delta > 0$ (indépendant du point). Il s'agit là d'une condition forte de *convexité*[10].

On admet ici le résultat suivant, validant (sous une hypothèse de convexité portant sur f) l'intérêt de la méthode algorithmique de descente que nous venons de présenter.

Proposition 3.11 (convergence de la méthode du gradient sous une hypothèse de convexité). *Si $f : \mathbb{R}^n \to \mathbb{R}$ (de classe C^2) est fortement convexe, l'algorithme du gradient avec pas optimal conduit, quelque soit le point d'où il est initié, à une suite convergeant vers une (en fait la, car elle est unique) solution optimale. Lorsque f est seulement convexe mais tend vers $+\infty$ lorsque $\|x\|$ tend vers l'infini, alors, quitte à extraire une sous suite, toute suite $(x_k)_{k\geq 0}$ générée par l'algorithme du gradient à pas optimal est minimisante.*

Exemple 3.8 (résolution d'un système linéaire). Si A est une matrice réelle (n,n) symétrique définie positive et B un vecteur de $\mathbb{R}^n$, la fonction
$$f_{A,B} : x \longmapsto \frac{{}^t xAx}{2} - \langle B\,,\, x\rangle$$
est fortement convexe. Le gradient de cette application $f_{A,B}$ est
$$\nabla f_{A,B} = Ax - B.$$

10. Une fonction $f : \mathbb{R}^n \to \mathbb{R}$ est convexe si et seulement si $f(tx+(1-t)y) \leq tf(x)+(1-t)f(y)$ pour tout x, y dans $\mathbb{R}^n$, pour tout $t \in [0,1]$. Dire que f est convexe équivaut à dire que la matrice H_f est en tout point une matrice symétrique positive (*i.e* de valeurs propres positives ou nulles).

Chercher l'unique point où f atteint son mimimum revient donc à résoudre le système linéaire $Ax = B$. Dans ce cas particulier, la recherche du pas optimal p (ligne 5 de l'algorithme 2 du gradient à pas optimal) est facilitée par le fait que

$$f(x - t\nabla f(x)) = f(x) + \frac{t^2}{2}\langle A.\nabla f(x)\,,\,\nabla f(x)\rangle - t\,\|\nabla f(x)\|^2$$

(ce que l'on voit en faisant le calcul). Le minimum (d'ailleurs global sur $\mathbb{R}$) de la fonction

$$t \mapsto f(x - t\nabla f(x))$$

est atteint en

$$p = \frac{\|\nabla f(x)\|^2}{\langle A.\nabla f(x), \nabla f(x)\rangle}.$$

et l'algorithme du gradient à pas optimal se résume dans ce cas à

$$x - \frac{\|\nabla f(x)\|^2}{\langle A.\nabla f(x), \nabla f(x)\rangle}\nabla f(x) \longleftarrow x. \tag{3.35}$$

Cette méthode est cependant loin d'être performante. Elle soutend cependant l'algorithme de descente dit *du gradient conjugué* présenté dans la section suivante.

3.4.5. Descente et gradient conjugué. On considère une fonction $f : \mathbb{R}^n \to \mathbb{R}$ dont le prototype sera ultérieurement la fonction

$$f_{A,B} : x \longmapsto \frac{{}^t xAx}{2} - \langle B\,,\,x\rangle$$

(où A est une matrice (n,n) réelle symétrique définie positive, B un vecteur de $\mathbb{R}^n$) introduite dans l'exemple 3.8. L'algorithme de descente est initié en x_0, mais cette fois les directions successives de descente $d_1, d_2, ...$ sont définies de manière interactive (on ne les choisit plus systématiquement suivant la règle $d_k = -\nabla f(x_k)$). Au lieu de cela, on essaye de choisir de proche en proche à la fois les directions d_k (par direction, on entend un vecteur non nul, mais auquel on n'impose pas d'etre unitaire) et les scalaires p_k, pour que, pour chaque k :

$$x_{k+1} = x_k + p_{k+1}\,d_{k+1}, \quad f(x_{k+1}) = \min_{x_0 + \mathrm{Vec}(d_1,\ldots,d_{k+1})} f.$$

Ce processus est en général impossible à réaliser (le minimum à droite porte sur un ensemble dépendant de k degrés de liberté, alors que l'on ne dispose que d'un seul paramètre p pour le réaliser) sauf si d_{k+1} est judicieusement choisie en fonction des précédentes.

Ceci est néanmoins possible à réaliser dans le cas particulier où $f = f_{A,B}$. Les choses se simplifient en effet dans ce cas particulier si l'on impose à Ad_{k+1} d'être orthogonal au sous espace engendré par $d_1, ..., d_k$ (ou encore, ce qui revient au même puisque A est symétrique, à faire en sorte que d_{k+1} soit orthogonal au sous-espace engendré par $Ad_1, ..., Ad_k$). En effet, dans ce cas, c'est le vecteur

$$x_{k+1} = x_k + \frac{\langle B - Ax_k\,,\,d_{k+1}\rangle}{\langle Ad_{k+1}\,,\,d_{k+1}\rangle}\,d_{k+1}$$

qui réalise, tout en étant de la forme $x_k + pd_{k+1}$ exigée, le minimum de f sur le sous espace $\mathrm{Vec}(d_1, ..., d_{k+1})$ (on prend ici $x_0 = 0$). Comme l'objectif est ici la résolution

Algorithme 4 la méthode du gradient conjugué

$k = 0\,;\, x_0 = 0\,;\, r_0 = B$
tant que $r_k \neq 0$ **faire**
  $k = k + 1$
  si $k = 1$ **alors**
    $d_1 = r_0$
  sinon
    Prendre p_k qui minimise $||d - r_{k-1}||$ sur Vect $(Ad_1, \ldots, Ad_{k-1})^{\perp}$
  fin si
  Prendre $\alpha_k = \dfrac{\langle d_k, r_{k-1}\rangle}{\langle d_k, Ad_k\rangle}$
  $x_k = x_{k-1} + \alpha_k d_k$
  $r_k = B - Ax_k$
fin tant que

approchée du système linéaire $Ax = B$, il est raisonnable de choisir la direction d_{k+1} (une fois $d_1, ..., d_k$ construites) de manière à ce que

$$\|d_{k+1} - (B - Ax_k)\| = \min_{d \in \mathrm{Vec}(Ad_1,\ldots,Ad_k)^{\perp}} \|d - (B - Ax_k)\|.$$

On génère ainsi l'algorithme 3 de *gradient conjugué* (ici dans sa version primitive).

3.4.6. Optimisation sous contraintes linéaires. On considère ici une fonction objectif $f\ : \mathbb{R}^n \to \mathbb{R}$ et un jeu de contraintes affines. Nous supposerons ici que ces contraintes sont du type « égalités-inégalités » affines, plus précisément de la forme :

$$\begin{aligned} &h_1(x) = \cdots = h_\mu(x) = 0 \\ &g_1(x) \leq 0, ..., g_\nu(x) \leq 0, \end{aligned} \tag{3.36}$$

les fonctions $h_1, ..., h_\mu, g_1, ..., g_\nu$ $(\mu + \nu = p \leq n)$ étant toutes ici des fonctions affines, *i.e* de la forme $\langle \alpha, x\rangle + \beta$, $\alpha \in \mathbb{R}^n \setminus \{0\}$, $\beta \in \mathbb{R}$. Cette situation est la plus courante dans les problèmes rencontrés dans la pratique (économie, recherche opérationnelle,...). On se place ici dans le cas particulier où les formes linéaires sous-jacentes aux formes affines $h_1, ..., h_\mu, g_1, ..., g_\nu$ sont linéairement indépendantes.

Nous allons ici nous contenter de décrire deux méthodes, la *méthode du gradient projeté*, directement inspirée de la méthode du gradient avec pas optimal (mais en général difficile à mettre en œuvre), puis la *méthode d'Uzawa*. Notons que lorsque $\mu = p \leq n$ (toutes les contraintes sont du type « égalité »), nous avons déjà signalé (Sous-Section 3.4.3) la possibilité d'attaquer le problème grâce à la méthode de Newton dans $\mathbb{R}^{n+p}$ appliquée au Lagrangien

$$L_{f;h}\ : (x, u^*) \longmapsto f(x) + \sum_{j=1}^{p} u_j^* h_j(x)$$

dont on tente d'approcher les zéros du gradient (on adopte ici la notation u^* pour les variables « duales »).

A. *La méthode du gradient projeté*

C'est la méthode la plus naturelle, directement inspirée par la méthode du gradient. Le sous-ensemble K de $\mathbb{R}^n$ défini par les contraintes (3.36) est un sous-ensemble

convexe fermé de $\mathbb{R}^n$, pour lequel on dispose d'un opérateur géométrique de projection

$$P_K \ : x \in \mathbb{R}^n \mapsto P_K[x] \in K,$$

où $P_K[x]$ est par définition l'unique point $y \in K$ tel que

$$\forall\, z \in K\,,\ \langle x - z, x - y\rangle \leq 0,$$

ou encore (ce qui est équivalent)

$$\|x - y\| = \inf_{z \in K} \|x - z\|.$$

Le point délicat est que cet opérateur est en général difficile à exprimer ; dans le cas particulier cependant où les contraintes sont toutes du type égalité (K est alors un sous-espace affine de $\mathbb{R}^n$), cet opérateur P_K est la projection orthogonale usuelle sur le sous-espace affine K (par exemple une droite, un plan affine,...) et se calcule donc aisément.

Voici l'algorithme sur lequel se fonde la méthode du gradient projeté (un pas maximal $p_{\max}$ et un seuil ϵ étant fixés) :

- On initie la méthode en un point x_0 de K.
- On choisit $0 < p_1 \leq p_{\max}$ judicieusement [11]. On peut choisir pour p_1 le p correspondant au pas optimal dans la méthode du gradient à pas optimal sans contraintes, mais ce choix (faisant fi des contraintes) ne s'avère souvent pas le plus adéquat.
- On calcule $y_1 = x_0 - p_1\, \nabla f\,(x_0)$. Ce point sort bien sûr du sous-ensemble K matérialisé par les contraintes, ce qui est un point qu'il va falloir corriger.
- On projette (c'est là l'étape la plus difficile) le point x_1 sur K :

$$x_1 = \mathrm{Pr}_K[y_1] = \mathrm{Pr}_K\Big[x_0 - p_1 \nabla f\,(x_0)\Big].$$

- On recommence avec x_1 en place de x_0, ce à condition toutefois que l'on ait $\|x_1 - x_0\| > \epsilon$ (sinon on s'arrête), puis on poursuit de la sorte l'algorithme.

La règle inductive est donc

$$x_{k+1} = \mathrm{Pr}_K\Big[x_k - p_{k+1}\, \nabla f\,(x_k)\Big],$$

le test d'arrêt étant

$$\|x_{k+1} - x_k\| \leq \epsilon.$$

Le synopsis de cet algorithme simple est présenté comme l'algorithme 5.

B. *La méthode d'Uzawa* [12]

On suppose toujours ici la fonction objectif $f \ : \mathbb{R}^n \to \mathbb{R}$ de classe C^1 et que les contraintes sont $h_1 = \cdots = h_\mu = 0, g_1 \leq 0, ..., g_\nu \leq 0$, où $h_1, ..., h_\mu, g_1, ..., g_\nu$, sont des fonctions affines correspondant à $p = \mu + \nu \leq n$ formes linéaires indépendantes sur $\mathbb{R}^n$.

---

11. Par exemple, si l'on sait *a priori* que f est fortement convexe et que

$$\langle \nabla f(x) - \nabla f(y), x - y\rangle \geq \delta \|x - y\|^2 \ \&\ \|\nabla f(x) - \nabla f(y)\| \leq M\, \|x - y\|\,,\ \forall\, x, y \in \mathbb{R}^n,$$

on veillera à choisir p dans un intervalle $[p_1, p_2]$ avec $0 < p_1 < p_2 < 2\delta/M$.

12. Initiée dans les années 1955 par Hirofumi Uzawa, mathématicien et économiste japonais (1928 –).

Algorithme 5 la méthode du gradient projeté

1: $k = 0\,;\, x_0 \in K\,;$
2: **tant que** $\|x_{k+1} - x_k\| \leq \epsilon$ **faire**
3: $k = k + 1$
4: Choisir un pas $p_k \leq p_{\max}$, par exemple celui correspondant au choix du pas optimal dans l'algorithme du gradient, *i.e*

$$f(p_k) = \inf_{t \in [0, p_{\max}]} f(x_k - t\,\nabla f\,(x_k)),$$

ou choisir un pas $p_k = p \leq p_{\max}$ constant.
5: $x_{k+1} = P_K[x_k - p_k\,\nabla f\,(x_k)]$
6: **fin tant que**

On introduit encore le Lagrangien $L_{f;h,g}$:

$$L_{f;h,g}\ : (x, u^*, v^*) \longmapsto f(x) + \sum_{j=1}^{\mu} u_j^* h_j(x) + \sum_{j=1}^{\nu} v_j^* g_j(x).$$

Si le point ξ réalise un minimum local de f sous les contraintes (3.36), on admettra qu'il existe qu'il existe des vecteurs $u_\xi^* \in \mathbb{R}^\mu$, $v_\xi^* \in \mathbb{R}^\nu$ tels que :

– on ait la *condition de stationnarité*

$$L_{f;h,g}(\xi, u_\xi^*, v_\xi^*) = \nabla f\,(\xi) + \sum_{j=1}^{\mu} u_{\xi,j}^*\,\nabla h_j\,(\xi) + \sum_{j=1}^{\nu} v_{\xi,j}^*\,\nabla g_j\,(\xi) = 0\ ; \tag{3.37}$$

– sont satisfaites les *conditions de faisabilité duale* :

$$\forall\, j = 1, ..., \nu,\ v_{\xi,j}^* \geq 0\ ; \tag{3.38}$$

– sont satisfaites enfin les *conditions de complémentarité* :

$$\forall\, j = 1, ..., \nu,\ v_{\xi,j}^*\, g_j(\xi) = 0. \tag{3.39}$$

Ces conditions toutes réunies (*stationnarité*, *faisabilité duale*, *complémentarité*) sont dites *conditions de Karush-Kuhn-Tucker* (dites KKT).

Si de plus la fonction f est supposée convexe, dire que ξ (vérifiant les contraintes $h = 0$, $g \leq 0$) est un minimum global de f sous les contraintes (3.36) équivaut à dire qu'il existe un couple $(u_\xi^*, v_\xi^*) \in \mathbb{R}^\mu \times \mathbb{R}^\nu$ tel que les conditions de Karush-Kuhn-Tucker (3.37), (3.38), (3.39) soient toutes les trois remplies. En effet, la convexité de f, le fait que les h_j et les g_j soient toutes des fonctions affines, font que la condition de stationnarité (3.37) implique que ξ est un minimum global de la fonction

$$x \mapsto F_\xi(x) = f(x) + \sum_{j=1}^{\mu} u_{\xi,j}^*\, h_j(x) + \sum_{j=1}^{\nu} v_{\xi,j}^*\, g_j(x).$$

Grâce aux conditions de complémentarité (3.39), il en résulte

$$\Big(h(x) = 0\ \&\ g_1(x) \leq 0, \ldots, g_\nu(x) \leq 0\Big) \Longrightarrow f(\xi) = F_\xi(\xi) \leq F_\xi(x) =$$

$$= f(x) + \sum_{j=1}^{\mu} u_{\xi,j}^*\, h_j(x) + \sum_{j=1}^{\nu} v_{\xi,j}^*\, g_j(x) \leq f(x),$$

ce qui exprime bien que ξ est un point où la fonction objectif f réalise son minimum global.

Une autre manière d'interpréter (toujours sous l'hypothèse que la fonction objectif f est C^1 et convexe et que les fonctions g_j et h_j soient affines) le fait que ξ soit un minimum global de f sous les contraintes (3.36) est de dire que l'on peut construire un point $(\xi, u^*(\xi), v^*(\xi)) \in \mathbb{R}^n \times \mathbb{R}^\mu \times [0,\infty[^\nu$ qui soit un *point selle* de $L_{f;h,g}$, *i.e* tel que l'on ait :

$$\begin{array}{l} \forall\, x \in \mathbb{R}^n,\ L_{f;h,g}(x, u^*_\xi, v^*_\xi) \geq L_{f;h,g}(\xi, u^*_\xi, v^*_\xi) \\ \forall\, u^* \in \mathbb{R}^\mu,\ \forall\, v^* \in [0,\infty[^\nu,\ L_{f;h,g}(\xi, u^*, v^*) \leq L_{f;h,g}(\xi, u^*_\xi, v^*_\xi). \end{array} \tag{3.40}$$

Algorithme 6 la méthode d'Uzawa : contraintes $h = 0$, $g \leq 0$

1: $k = 0$; $x_0 \in \mathbb{R}^n$; $u^{[0]\,*} \in \mathbb{R}^\mu$; $v^{[0]\,*} \in [0,\infty[^\nu$
2: **tant que** $\|x_{k+1} - x_k\| \leq \epsilon$ **faire**
3: $k = k + 1$
4: Déterminer un minimum absolu x_k solution du problème d'optimisation sans contraintes

$$f(x_k) = \inf_{x\in\mathbb{R}^n} \Big[f(x) + \sum_{j=1}^{\mu} u_j^{[k]\,*}\, h_j(x) + \sum_{j=1}^{\nu} v_j^{[k]\,*}\, g_j(x) \Big]$$

via (par exemple) la méthode du gradient à pas optimal (algorithme 3)
5: $u_j^{[k+1]\,*} = u_j^{[k]\,*} + p\, h_j(x_k)$, $j = 1, ..., \mu$
6: $v_j^{[k+1]\,*} = \max \Big[0, v_j^{[k]\,*} + p\, g_j(x_k)\Big]$, $j = 1, ..., \nu$
7: **fin tant que**

L'algorithme d'Uzawa est précisément fondé sur cette idée : la recherche d'un point ξ (satisfaisant aux contraintes) où la fonction objectif f présente (sous ces contraintes) un minimum absolu passe par la traque d'un point selle pour le Lagrangien :

$$(x, u^*, v^*) \in \mathbb{R}^n \times \mathbb{R}^\mu \times [0,\infty[^\nu \longmapsto L(x) + \sum_{j=1}^{\mu} u_j^*\, h_j(x) + \sum_{j=1}^{\nu} v_j^*\, g_j(x).$$

On en propose (avec l'algorithme 6) une version à pas constant p (et seuil $\epsilon > 0$).

Remarque 3.9 (choix du pas constant p dans le cas où la fonction objectif est fortement convexe). Si la fonction objectif f est fortement convexe et satisfait donc $\langle \nabla f\,(x) - \nabla f\,(y)\rangle \geq \delta \|x - y\|^2$ avec un $\delta > 0$, et si l'application linéaire sous-jacente à (g, h) $: \mathbb{R}^n \to \mathbb{R}^p$ est de norme inférieure ou égale à C, il convient de choisir $p \leq 2\delta/C^2$ pour que l'algorithme d'Uzawa conduise à une approximation du point ξ correspondant au minimum de la fonction objectif f. Cette situation se produit dans le cas particulier où

$$f(x) = f_{A,B}(x) = \frac{{}^t x A x}{2} - \langle B, x\rangle,$$

où A est une matrice symétrique (n, n) réelle définie positive et B un vecteur de $\mathbb{R}^n$ (la fonction objectif est *quadratique*). Dans ce cas, on peut choisir δ comme

$$\delta = \inf |\text{valeurs propres}\,(A)|.$$

CHAPITRE 4

Interpolation, approximation, modélisation

4.1. Les fonctions *spline* en 1D

On considère un maillage

$$x_0 = a < x_1 < x_2 < \cdots < x_N \leq x_{N+1} = b.$$

L'objectif que l'on se fixe ici est d'interpoler des valeurs réelles imposées $y_1, ..., y_N$ aux points $x_1, ..., x_N$, *i.e* de réaliser avec une fonction $s : [a, b] \to \mathbb{R}$ les conditions $s(x_j) = y_j$, $j = 1, ..., N$. Par contre, ce que l'on souhaite maintenant « minimiser »[1] est l'*énergie de flexion* du graphe de s (ce graphe se matérialisait autrefois par une règle flexible d'épaisseur infinitésimalement petite). Comme la courbure (en valeur absolue) au point courant $(x, s(x))$ est donnée par

$$\rho(x) = \frac{|s''(x)|}{(1 + |s'(x)|^2)^{3/2}},$$

ce que l'on cherche à minimiser pour construire la fonction s (parmi les fonctions candidates de classe C^2 sur $[a, b]$) est

$$I = \int_a^b \frac{|s''(t)|^2}{(1 + |s'(t)|^2)^3}\, dt.$$

Si l'on suppose que les variations de s sont limitées (ce qu'aussi on impose), ceci se ramène à tenter de minimiser

$$\widetilde{I} = \int_a^b |s''(t)|^2\, dt \simeq I.$$

La fonction s réalisant cet objectif de minimisation doit obéir aux exigences suivantes :

- être de classe C^2 sur $[a, b]$;
- être affine sur les deux intervalles extrèmes $[a, x_1]$ et $[x_N, b]$;
- coincider avec un polynôme de degré 3 (la dérivée seconde est alors affine[2] sur chaque segment $[x_j, x_{j+1}]$, $j = 1, ..., N - 1$).

Nous venons de dégager ici le concept de 3-spline. Mais on peut imposer plus de régularité et réaliser la notion de $2q - 1$-*spline*.

DÉFINITION 4.1 (notion de spline de degré $2q - 1$). Soient $N \geq q \geq 1$ et

$$a = x_0 < x_1 < \cdots < x_N < x_{N+1} = b \tag{4.1}$$

1. On se différentie ici de ce qui se passait lors de l'interpolation de Lagrange, ou pareil souci était absent.

2. C'est logique compte tenu de l'exigence de minimisation : la ligne droite est le plus court chemin d'un point à un autre.

un maillage de $[a,b]$ à $N+1$ nœuds. Le $\mathbb{R}$-espace vectoriel des $2q-1$-*spline* subordonné à ce maillage est le $\mathbb{R}$-espace vectoriel des fonctions de classe C^{2q-2} sur $[a,b]$, qui sont polynomiales de degré $q-1$ sur les deux intervalles extrémaux $[a,x_1]$ et $[x_N,b]$ et polynomiales de degré au plus $2q-1$ sur chacun des segments internes $[x_j,x_{j+1}]$, $j=1,...,N-1$. Le $\mathbb{R}$-espace vectoriel des $(2q-1)$-spline est noté $\mathcal{S}_{2q-1}$ (on omet le fait qu'il soit relatif au maillage (4.1)).

EXEMPLE 4.2 (*spline* de degré 1). L'exemple $2\times 1-1=1$ des 1-*splines* est particulièrement important ; les fonctions sont continues, affines par morceaux sur chaque segment $[x_j,x_{j+1}]$, $j=1,...,N-1$, et constantes sur les deux intervalles extrêmes $[a,x_1]$ et $[x_N,b]$.

Le résultat majeur en direction de l'interpolation est le suivant :

PROPOSITION 4.1 (interpolation par les *q-spline*). *Si $N\geq q\geq 1$ et si un maillage* (4.1) *de $[a,b]$ est donné, il existe, étant donnés $y_1,...,y_N$ des réels fixés, un et un seul spline s de degré $2q-1$ interpolant les valeurs y_j aux points x_j, $j=1,...,N$. Le $\mathbb{R}$-espace des $(2q-1)$-spline associés au maillage* (4.1) *est donc de dimension N.*

DÉMONSTRATION. Ceci tient au fait que tout *spline* de degré $(2q-1)$ subordonné au maillage (4.1) s'exprime

$$s(x)=p_{\text{init}}(x)+\sum_{j=1}^{N}\gamma_j\frac{(x-x_j)_+^{2q-1}}{(2q-1)!}$$

où $(x-x_j)_+:=\max(x-x_j,0)$ et $p_{\text{init}}(x)$ est la fonction polynomiale de degré $q-1$ restriction de s à $[a,x_1]$. Le réel γ_j, $j=1,...,N$, est la valeur du saut de discontinuité de la dérivée $s^{(2q-1)}$ au point x_j. On trouve les conditions

$$\sum_{j=1}^{N}\gamma_j x_j^k=0,\ 0\leq k\leq q-1 \tag{4.2}$$

en écrivant que la restriction de s à $[x_N,b]$, qui est la fonction polynomiale

$$x\mapsto p_{\text{fin}}(x)=p_{\text{init}}(x)+\sum_{j=1}^{N}\gamma_j\frac{(x-x_j)^{2q-1}}{(2q-1)!},$$

doit être une fonction polynomiale de degré $q-1$, ce qui est réalisé si on écrit l'annulation en zéro des dérivées de $s^{(q)}$ à l'ordre $0,1,...,q-1$; on obtient ainsi les conditions (4.2). En ajoutant les q conditions (4.2) aux N conditions d'interpolation

$$s(x_l)=p_{\text{init}}(x_l)+\sum_{j=1}^{l}\gamma_j\frac{(x_l-x_j)^{2q-1}}{(2q-1)!}=y_l,\ l=1,...,N,$$

on trouve $N+q$ conditions pour $N+q$ paramètres à déterminer (les coefficients de p_{init} et les γ_j, ce qui fait bien le compte). Reste à montrer que le système ainsi posé est bien de Cramer, c'est-à-dire que le système homogène admet comme seule solution la solution nulle. Ceci tient à un remarque fort utile : si l'on introduit le $\mathbb{R}$-espace $\mathcal{H}^q$ (relatif au maillage (4.1), ce que pour simplifier on omet de faire apparaitre) des fonctions de classe C^{q-1} sur $[a,b]$, telles que la restriction à chaque

$]x_j, x_{j+1}[$, $j = 0, N$, se prolonge en une fonction de classe C^q sur $[x_j, x_{j+1}]$, alors, pour tout $f \in \mathcal{H}^q$, on a, si

$$s = p_{\text{init}}(x) + \sum_{j=1}^{N} \gamma_j \frac{(x - x_j)_+^{2q-1}}{(2q-1)!} \in \mathcal{S}_{2q-1},$$

la relation

$$\int_a^b s^{(q)}(t) f^{(q)}(t)\, dt = (-1)^q \sum_{j=1}^{N} \gamma_j f(x_j) \tag{4.3}$$

(qui se prouve *via* une intégration par parties, voir par exemple [**MathAp**], chapitre 2. En particulier, si $f = s$ et $s(x_j) = 0$, $j = 1, ..., N$, il vient

$$\int_a^b |s^{(q)}(t)|^2\, dt = 0,$$

ce qui prouve que s est un polynôme de degré $q-1$, nul en $N \geq q$ points, donc identiquement nul ; les coefficients γ_j et le polynôme p_{init} sont identiquement nuls. Le système homogène

$$\begin{aligned} &p_{\text{init}}(x_k) + \sum_{j=1}^{l} \gamma_j \frac{(x_l - x_j)^{2q-1}}{(2q-1)!} = 0,\ l = 1, ..., N, \\ &\sum_{j=1}^{N} \gamma_j x_j^k = 0,\ 0 \leq k \leq q-1 \end{aligned} \tag{4.4}$$

n'admet donc que la solution nulle et le système

$$\begin{aligned} &p_{\text{init}}(x_k) + \sum_{j=1}^{l} \gamma_j \frac{(x_l - x_j)^{2q-1}}{(2q-1)!} = y_l,\ l = 1, ..., N, \\ &\sum_{j=1}^{N} \gamma_j x_j^k = 0,\ 0 \leq k \leq q-1 \end{aligned} \tag{4.5}$$

est bien de Cramer. □

Remarque 4.3. Le calcul du *spline* s réalisant l'interpolation implique la résolution d'un système linéaire en général mal conditionné. On peut toutefois ramener ce problème à la résolution d'un système « creux » (la matrice du système se présentant comme une matrice bande autour de la première diagonale). Ce type de calcul pourra être envisagé en TD et en TP.

Remarque 4.4. Parmi toutes les fonctions $g \in \mathcal{H}^q$ (sous-entendu, relatif au maillage (4.1)) interpolant aussi les valeurs y_j aux points x_j, $j = 1, ..., N$, le *spline*

s de degré $(2q-1)$ que l'on vient de trouver est tel que

$$\begin{aligned}\int_a^b |g^{(q)}(t)|^2\,dt &= \int_a^b |s^{(q)}(t)|^2\,dt + \int_a^b |(g-s)^{(q)}(t)|^2\,dt + 2\int_a^b s^{(q)}(t)g^{(q)}(t)\,dt \\ &= \int_a^b |s^{(q)}(t)|^2\,dt + \int_a^b |(g-s)^{(q)}(t)|^2\,dt \\ &\qquad +2(-1)^q \sum_{j=1}^N \gamma_j\,[g-s](x_j) \\ &= \int_a^b |s^{(q)}(t)|^2\,dt + \int_a^b |(g-s)^{(q)}(t)|^2\,dt + 0\end{aligned}$$

(d'après la relation (4.3), si les γ_j sont ceux associés au *spline* s), ce qui prouve que

$$\int_a^b |s^{(q)}(t)|^2\,dt$$

est bien minimal parmi les $\int_a^b |g^{(q)}(t)|^2\,dt$ lorsque g parcourt $\mathcal{H}^q$ (relatif au maillage). Ceci est bien en phase avec le souci de minimisation de l'énergie de flexion dans le choix $q=2$ conduisant à la construction des $2\times 2-1=3$-*splines*.

4.2. Approximation polynomiale sur un intervalle de $\mathbb{R}$

Étant donnée une fonction continue sur un intervalle (a,b) (bornes incluses ou non) $-\infty \le a < b \le +\infty$ de $\mathbb{R}$, à valeurs réelles et complexes, une autre façon d'esquiver les difficultés sous jacentes à l'interpolation de Lagrange (que celle introduite *via* les fonctions *spline*, voir la section 4.1), tout en approchant f par des fonctions mobilisables algorithmiquement (par exemple les fonctions polynomiales), est d'envisager l'approximation de f par des fonctions polynomiales de degré imposé dans le $\mathbb{R}$ ou $\mathbb{C}$-espace vectoriel des fonctions de (a,b) dans $\mathbb{R}$ ou $\mathbb{C}$, équipé d'une norme : les normes les plus intéressantes seront pour nous ici de deux types :

(1) la norme uniforme $\|\ \|_\infty$ définie par $\|f\|_\infty := \sup_{(a,b)} |f(t)|$, le $\mathbb{R}$ ou $\mathbb{C}$-espace vectoriel normé $C([a,b],\mathbb{R}$ ou $\mathbb{C})$ étant complet (pour cette norme) lorsque (a,b) est fermé ;

(2) la norme $L^2(\mathbb{R},\omega(t)\,dt)$ définie par

$$\|f\|_{2,\omega}^2 := \int_{(a,b)} |f(t)|^2\,\omega(t)\,dt, \tag{4.6}$$

où ω désigne une fonction positive, finie presque partout sur (a,b), mesurable au sens de Lebesgue (on dit aussi un « poids ») telle que

$$\forall\, n\in\mathbb{N}\,,\qquad \int_{(a,b)} |t|^n\,\omega(t)\,dt < +\infty\ ; \tag{4.7}$$

cette norme est particulièrement importante car il lui est attaché une notion d'*orthogonalité* ; le produit scalaire correspondant est défini par

$$\langle f,g\rangle_\omega := \int_{(a,b)} f(t)\,\overline{g(t)}\,\omega(t)\,dt, \tag{4.8}$$

ce produit scalaire étant défini (inégalité de Cauchy-Schwarz) pourvu que f et g soient toutes les deux d'énergie finie relativement au *poids* ω, *i.e*

$$\int_{(a,b)} |f(t)|^2\,\omega(t)\,dt < \infty \quad \& \quad \int_{(a,b)} |g(t)|^2\,\omega(t)\,dt < +\infty\ ;$$

du point de vue physique, la quantité $\langle f, g\rangle_\omega$ s'interprète en termes de *corrélation*, tandis que $\|f\|^2_{2,\omega}$ correspond à une notion d'*énergie*; notons toutefois que dans ce cas le $\mathbb{R}$ ou $\mathbb{C}$-espace vectoriel $C([a,b], \mathbb{R}$ ou $\mathbb{C})$ n'est plus complet pour la norme $\|\ \|_\omega$, son complété étant l'espace de Hilbert $L^2((a,b), \mathcal{B}(\mathbb{R}), \omega(t)\,dt))$.

4.2.1. Approximation en norme infinie sur un segment $[a,b]$. Nous introduisons dans un premier temps un puissant outil d'approximation polynomiale en norme uniforme, les *polynômes de Bernstein*. Notons que nous ne nous préoccupons pas pour l'instant de réaliser la « meilleure » approximation polynomiale (à degré imposé) en norme uniforme, mais de voir comment choisir le degré assez grand pour réaliser une approximation uniforme avec un seuil de tolérance d'erreur (ici ϵ) imposé (*i.e.* $\sup_{[a,b]} |f - f_{\text{app}}| \leq \epsilon$).

Définition 4.5 (polynômes de Bernstein d'une fonction continue). Soit f une fonction continue sur le segment $[0,1]$ de $\mathbb{R}$, à valeurs réelles ou complexes et $n \in \mathbb{N}^*$. Le *n-ième polynôme de Bernstein* $B_n[f]$ de f sur $[0,1]$ est défini [3] par

$$B_n[f](X) := \sum_{k=0}^{n} f(k/n) \binom{n}{k} X^k (1-X)^{n-k}. \tag{4.9}$$

Le résultat majeur que soutend cette définition est le résultat suivant.

Theorème 4.6 (théorème d'approximation uniforme de Bernstein). *Si f est une fonction continue sur $[0,1]$ et que $\delta(\epsilon)$ est suffisamment petit pour que*

$$\forall\, t, s \in [0,1],\ |t-s| \leq \delta \Longrightarrow |f(t) - f(s)| \leq \epsilon,$$

on a

$$\forall\, t \in [0,1],\ |f(t) - B_n[f](t)| \leq 2\epsilon \tag{4.10}$$

pourvu que

$$n \geq 1 + \frac{\|f\|_\infty}{2\epsilon\delta^2}.$$

Démonstration. On majore, pour $t \in [0,1]$, $|f(t) - B_n[f](t)|$ par

$$\begin{aligned} &\sum_{k=0}^{n} \epsilon \binom{n}{k} t^k (1-t)^{n-k} + \sum_{\{0\leq k\leq n\,;\,|t-k/n|>\delta\}} |f(t) - f(k/n)| \binom{n}{k} t^k (1-t)^{n-k} \\ \leq\ & \epsilon + \frac{2\|f\|_\infty}{n\delta^2}\, t(1-t) \leq \epsilon + \frac{\|f\|_\infty}{2n\delta^2} \end{aligned}$$

3. Noter que, pour tout $p \in [0,1]$, la suite des nombres

$$\binom{n}{k} p^k (1-p)^{n-k},\ k = 0, ..., n, \tag{$*$}$$

représente la loi binomiale $\mathcal{B}(n,p)$, ce qui explique qu'il faille interpréter $B_n[f](t)$, pour $t \in [0,1]$, comme une « valeur moyenne » des $f(k/n)$ pondérés par les coefficients $(*)$. On retrouvera cette idée (exploitée cette fois géométriquement) dans le tracé des *courbes de Bézier* à partir de points de contrôle $M_0, ..., M_n$ du plan ou de l'espace.

en utilisant les relations

$$\sum_{k=0}^{n} \binom{n}{k} t^k (1-t)^{n-k} = 1$$

$$\sum_{k=0}^{n} (nt-k)^2 \binom{n}{k} t^k (1-t)^{n-k} = nt(1-t)$$

valables pour tout $t \in [0,1]$ (penser par exemple au calcul de la variance $p(1-p)$ de la loi binomiale de paramètre $p \in [0,1]$) et à l'inégalité $t(1-t) \leq 1/4$ sur $[0,1]$. □

REMARQUE 4.7. Lorsque $[0,1]$ est remplacé par le segment $[a,b]$, c'est le changement de variable $t \leftrightarrow a + u(b-a)$ qui permet de ramener le problème de l'approximation polynomiale uniforme sur $[a,b]$ à celui de l'approximation polynomiale uniforme sur $[0,1]$.

4.2.2. Polynômes de Bernstein et courbes de Bézier. Le tracé de courbes planes joignant des points (dits de *contrôle*) du plan de manière « régulière » (voir par exemples les logiciels de dessin graphique tels `xfig`, etc.) s'inspire du procédé d'extrapolation/interpolation qui a conduit dans la section précédente à la construction du polynôme de Bernstein $B_n[f]$ à partir des valeurs de f aux nœuds d'un maillage régulier du segment $[a,b]$.

DÉFINITION 4.8 (courbe de Bézier). Étant donnés $n+1$ points $M_0, ..., M_n$ du plan (repéré par rapport à un repère orthonormé $(0; \vec{i}, \vec{j})$), on appelle courbe paramétrée de Bézier [4] associée à ces $n+1$ points (dits alors « de contrôle ») pris dans l'ordre [5] requis $M_0, ..., M_n$, la courbe paramétrée :

$$t \in [0,1] \longmapsto B[M_0, ..., M_n](t) := \sum_{k=0}^{n} \binom{n}{k} t^k (1-t)^{n-k} M_k. \tag{4.11}$$

Du point de vue géométrique, les points du support de la courbe de Bézier

$$t \mapsto B[M_0, ..., M_n](t)$$

(c'est-à-dire l'image de $[0,1]$ par cette courbe) sont toujours dans l'enveloppe convexe de l'ensemble fini $\{M_0, ..., M_n\}$ et l'arc paramétré $t \mapsto B[M_0, ..., M_n](t)$ est tangent au segment $[M_0, M_1]$ au point M_0 $(t=0)$ et au segment $[M_{n-1}, M_n]$ au point M_n $(t=1)$.

La construction algorithmique (récursive) de la courbe de Bézier $t \mapsto B[M_0, ..., M_n](t)$ à partir de $n+1$ points de contrôle se fonde sur la formule immédiate :

$$B_n[M_0, ..., M_n](t) = (1-t)B_{n-1}[M_0, ..., M_{n-1}](t) + tB_{n-1}[M_1, ..., M_n](t). \tag{4.12}$$

On retrouve à partir de cette formule une démarche algorithmique « en triangle » en tout point semblable à celle que l'on a mis en œuvre pour construire les différences divisées successives (section 3.1.2) ou fabriquer le polynôme d'interpolation de Lagrange suivant le procédé guidé par le lemme d'Aitken (lemme 2.12). C'est l'algorithme de *De Casteljau* [6], que l'on pourra implémenter sur des exemples et qui

4. Le concept est récent : on le doit à l'ingénieur (mécanicien, électricien) français (chez Renault) Pierre Bézier (1910-1999).

5. L'ordre joue ici un rôle important !

6. Paul de Faget de Calteljau est un ingénieur français contemporain (1930–) ; c'est comme ingénieur chez Citroën qu'il développa cet algorithme.

fournit une aide dirigée commode au tracé graphique (dessin industriel, *design*, *etc.*).

4.2.3. Meilleure approximation polynomiale uniforme sur un segment à degré prescrit. Si le segment $[a,b]$ est donné et que $f : [a,b] \mapsto \mathbb{R}$ ou $\mathbb{C}$ est une fonction continue sur $[a,b]$, on peut naturellement se demander, $n \geq 1$ étant un entier fixé, quelle fonction polyomiale $p_n[f]$ (de degré au plus n) réalise la distance de f au sous-espace vectoriel (de dimension $n+1$) engendré par les fonctions monomiales $t \mapsto t^k$, $k = 0, ..., n$.

Cette question est ici délicate car nous ne sommes pas en présence d'une norme (en l'occurrence ici la norme $\| \ \|_\infty = \sup_{[a,b]} | \ |$) dérivant d'un produit scalaire, donc se prêtant aux raisonnements géométriques inspirés du théorème de Pythagore (comme nous le ferons plus loin avec les normes $\| \ \|_{2,\omega}$).

Il est naturel de penser qu'une fonction polynomiale de degré n réalisant la meilleure approximation se doive de se présenter comme une fonction « oscillante » autour de f. Nous énonçons ceci comme un résultat admis, fondement de la théorie de l'approximation développée par P. Tchebychev [7] dans le cadre de l'approximation uniforme des fonctions continues à valeurs réelles sur un segment $[a,b]$ de $\mathbb{R}$.

THEORÈME 4.9 (théorème d'alternance de Tchebychev). *Si f est une fonction continue dur $[a,b]$, à valeurs réelles, dire que le polynôme p_n de degré au plus n réalise la distance uniforme de f au $\mathbb{R}$-sous-espace $(\mathbb{R}_n[t])_{|[a,b]}$ des restrictions à $[a,b]$ des fonctions polynomiales de degré au plus n équivaut à dire qu'il existe un maillage*

$$a \leq x_0 < x_1 < \cdots < x_{n-1} < x_n \leq b$$

tel que, en chaque nœud x_k, $k = 0, ..., n$, on ait

$$|(f-p)(x_k)| = \sup_{[a,b]} |f-p|,$$

ce de manière à ce que soient respectées, en ces nœuds, les alternances de signe :

$$\forall\, k = 0, ..., n, \ \ f(x_{k+1}) - p(x_{k+1}) = -(f(x_k) - p(x_k)).$$

EXEMPLE 4.10 (première apparition des polynômes de Tchebychev). Si $k \in \mathbb{N}$, on définit le polynôme de Tchebychev T_k par la relation trigonométrique

$$T_k(\cos\theta) = \cos(k\theta) \quad \forall\, \theta \in \mathbb{R}.$$

Ce polynôme est un polynôme de degré exactement k et l'on a

$$T_k(t) = 2^{k-1}\, t^k + \cdots$$

Si n est un entier positif donné, une meilleure approximation de $t \mapsto t^{n+1}$ sur $[-1,1]$ par un élément de $(\mathbb{R}_n[t])_{|[-1,1]}$ est donnée par la fonction polynomiale

$$t \mapsto p(t) = t^{n+1} - \frac{1}{2^n} T_{n+1}(t).$$

En effet, le polynôme T_{n+1} prend alternativement les valeurs 1 et -1 aux $n+2$ points

$$x_k = \cos\frac{k\pi}{n+1}, \quad k = 0, ..., n+1$$

7. Le mathématicien russe Pafnouti Tchebychev (1821-1894), outre ses travaux fondamentaux en théorie de l'approximations, est l'un des pères de ce qui deviendra la théorie moderne des probabilités.

de $[-1,1]$. Comme $|T_{n+1}(t)| \leq 1$ sur $[-1,1]$, la règle d'alternance de Tchebychev est bien satisfaite lorsque $f(t) = t^{n+1}$ et

$$p(t) = t^{n+1} - \frac{1}{2^n} T_{n+1}(t).$$

On vient donc d'exhiber, avec cette fonction polynomiale p, la meilleure approximation uniforme sur $[-1,1]$ de $t \mapsto t^{n+1}$ par un élément du sous-espace des restrictions à $[-1,1]$ des éléments de $\mathbb{R}_n[t]$. Cet exemple illustre le rôle très important joué par les polynômes de Tchebychev en ce qui concerne le problème de l'approximation polynomiale uniforme sur un segment de $\mathbb{R}$.

Si f est une fonction de classe C^m sur un segment $[a,b]$ de $\mathbb{R}$, à valeurs réelles, et si $\xi_0, ..., \xi_n$ sont n points de $[a,b]$, on rappelle (Proposition 3.4) que l'on a l'estimation

$$\begin{aligned} &\forall\, t \in [a,b],\ |f(t) - \text{Lagrange}\,[\xi_0, ..., \xi_n\ ; f(\xi_0), ..., f(\xi_n)](t)| \\ &\leq \frac{\sup_{[a,b]} |f^{(n+1)}(t)|}{(m+1)!} \Big| \prod_{j=0}^{n} (t - \xi_j) \Big| \\ &\leq \frac{\sup_{[a,b]} |f^{(n+1)}(t)|}{(m+1)!}\, |t^{n+1} - p_{n,\xi}(t)| \end{aligned} \tag{4.13}$$

Si p désigne un polynôme de meilleure approximation uniforme pour $t \mapsto t^{n+1}$ sur le $\mathbb{R}$-espace vectoriel $(\mathbb{R}_n[t])_{|[a,b]}$, on constate (avec le théorème d'alternance de Tchebychev et le théorème des valeurs intermédiaires) que la fonction polynomiale

$$t \mapsto t^{n+1} - p(t)$$

(de degré $n+1$) admet $n+1$ zéros $\theta_0, ..., \theta_n$ (donc de fait tous ses zéros) dans $[a,b]$. Le choix de ces points $\theta_0, ..., \theta_n$ en place des ξ_j, $j = 0, ..., n$, dans (4.13) minimise l'erreur uniforme commise en remplacant f sur $[a,b]$ par son polynôme d'interpolation de Lagrange aux points $\xi_0, ..., \xi_n$.

4.3. Approximation, modélisation et orthogonalité

4.3.1. Quelques familles importantes de polynômes orthogonaux. Si (a,b) est un intervalle de $\mathbb{R}$ et $\omega : (a,b) \to [0,\infty]$ une fonction positive (un *poids*) intégrable sur (a,b), la notion d'énergie « pondérée par le poids ω »

$$f \in C([a,b],\mathbb{C}) \mapsto \int_{(a,b)} |f(t)|^2\, \omega(t)\, dt$$

induit une notion d'orthogonalité (on se limitera ici au cadre des fonctions continues sur $[a,b]$) : deux fonctions f et g de $C((a,b),\mathbb{C})$, d'énergie finie relativement à ce poids, sont *orthogonales* ou *non corrélées* si et seulement si

$$\langle f, g \rangle_\omega = \int_{(a,b)} f(t)\, \overline{g(t)}\, \omega(t)\, dt = 0.$$

Les restrictions à (a,b) des fonctions polynomiales [8] (intéressantes pour nous car mobilisables du point de vue informatique) ne sont pas orthogonales relativement à ce poids, ce qui rend difficile la recherche, étant donné un entier n, la recherche de

8. On suppose ici que les restrictions de ces fonctions polynomiales sont toutes d'énergie finie relativement au poids, ce qui revient à supposer que pour tout entier n,

$$\int_{(a,b)} |t|^n\, \omega(t)\, dt < +\infty.$$

la fonction polynomiale de degré prescrit la « plus proche » d'une fonction continue $f : (a,b) \to \mathbb{C}$ au sens de la distance quadratique correspondant à l'énergie, *i.e*

$$d_\omega(f,g) = \sqrt{\int_{(a,b)} |f(t)-g(t)|^2\, \omega(t)\, dt}. \tag{4.14}$$

Nous donnons ici quatre exemples importants de tels espaces quadratiques à poids et quatre familles de fonctions polynomiales $(P_n)_{n\geq 0}$ en relation directe avec chacune d'elles.

Étant donné un tel poids $\omega : (a,b) \to [0,\infty]$, on appelle *famille de polynômes orthogonaux unitaires* $(P_n)_{n\geq 0}$ attachée à ce poids la famille de polynômes unitaires ($\deg P_n = n$, $P_n(t) = t^n + \cdots$) construite à partir du système libre $\{1, t, t^2, ..., t^k, ...\}$ des restrictions à (a,b) des fonctions monomiales suivant le procédé d'orthogonalisation de Gram-Schmidt [9].

Il résulte de cette construction que P_n a exactement n racines réelles, toutes simples, qui sont toutes contenues dans $]a,b[$. En effet, on peut écrire

$$P_n(t) = \prod_{\lambda=1}^{l_n} (t-\xi_\lambda) \times \prod_{\mu=1}^{k_n} (t-\eta_\mu) \times Q_n(t),$$

où $l_n + k_n \leq n$, $\xi_1, ..., \xi_{l_n}$ sont les zéros réels de P_n intérieurs appartenant à $]a,b[$ et de multiplicité impaire, $\eta_1, ..., \eta_k$ les autres zéros réels, et Q_n est une fonction polynomiale unitaire de degré $n - k_n - l_n$ ne s'annulant pas dans $\mathbb{R}$. On remarque, si $l_n < n$, que

$$\int_{(a,b)} P_n(t) \Big(\prod_{\lambda=1}^{l_n} (t-\xi_\lambda) \Big)\, \omega(t)\, dt = 0,$$

ce qui implique, puisque la fonction polynomiale

$$t \in]a,b[\longmapsto P_n(t) \prod_{\lambda=1}^{l_n} (t-\xi_\lambda) \tag{4.15}$$

ne saurait par construction changer de signe sur $]a,b[$, que cette fonction polynomiale (4.15) est identiquement nulle, ce qui est absurde. On a donc bien $l_n = n$ et les zéros de P_n sont tous réels, simples, et dans $]a,b[$. Pareil fait sera appelé à jouer un rôle important pour nous ultérieurement. Remarquons ici que les représentations graphiques des fonctions polynomiales $t \in]a,b[\mapsto P_n(t)$ se présenteront comme les représentations graphiques de fonctions de plus en plus oscillantes.

La construction de la famille des polynômes orthogonaux unitaires correspondant à un poids donné ω se fait suivant le schéma algorithmique 7.

Étant donnée une telle famille orthogonale unitaire $(P_n)_{n\geq 0}$ attachée à la notion d'orthogonalité sur $C((a,b),\mathbb{C})$ induite par le poids $\omega : (a,b) \to [0,\infty]$, il est facile de construire la meilleure approximation polynomiale (au sens de la distance quadratique (4.14)) de degré prescrit n pour une fonction continue (et d'énergie

---

On fera toujours cette hypothèse. Mieux, on supposera toujours que les restrictions à (a,b) de telles fonctions polynomiales forment un sous-espace dense dans le $\mathbb{C}$-espace vectoriel $C((a,b),\mathbb{C})$ des fonctions continues de (a,b) dans $\mathbb{C}$ pour cette distance quadratique d_ω.

9. Voir le cours d'Algorithmique Algébrique II, UE MHT631.

finie) $f : (a,b) \to \mathbb{C}$. Si les polynômes $(\widetilde{P}_n)_{n\geq 0}$ sont les polynômes P_n de la famille, chacun multiplié par la constante $\lambda_n = 1/\|P_n\|_\omega$ de manière à ce que

$$\int_{(a,b)} |\widetilde{P}_n(t)|^2 \, dt = 1,$$

la fonction polynomiale p de degré prescrit n telle que la distance

$$\sqrt{\int_{(a,b)} |f(t) - p(t)|^2 \, \omega(t) \, dt}$$

soit minimale (parmi toutes les fonctions polynomiales de degré au plus n) est la projection orthogonale sur le sous-espace engendré par les fonctions monomiales $\{1, t, t^2, ..., t^n\}$ ou, ce qui revient au même, la projection orthogonale sur le sous-espace $\{P_0, P_1, ..., P_n\}$, c'est-à-dire

$$p = \sum_{k=0}^{n} \Big(\int_{(a,b)} f(t) \, \overline{\widetilde{P}_k(t)} \, \omega(t) \, dt \Big) \, \widetilde{P}_k.$$

Algorithme 7 Génération des polynômes orthogonaux unitaires $P_0, P_1, ...$ relativement à $\omega : (a,b) \to [0,\infty]$

$P = [P_{-1} \equiv 0, P_0 \equiv 1]$
pour $k = 1$ **jusqu'à** n **faire**
  si $k = 1$ **alors**
    $\gamma = 0$
  sinon

$$\gamma = \frac{\langle P_{k-1}(t), P_{k-1}(t) \rangle_\omega}{\langle P_{k-2}(t), P_{k-2}(t) \rangle_\omega}$$

  fin si

$$c = \frac{\langle t\, P_{k-1}(t), P_{k-1}(t) \rangle_\omega}{\langle P_{k-1}(t), P_{k-1}(t) \rangle_\omega}$$

  $P_k(t) = (t - c)P_{k-1}(t) - \gamma P_{k-2}(t)$
  $P = [P, P_k]$
fin pour

a) *Les polynômes de Legendre*

On prend $(a,b) = [-1,1]$ (on peut ramener par changement de variables tout segment $[a,b]$ à ce cadre) et $\omega(t) \equiv 1$.

DÉFINITION 4.11. La famille des *polynômes de Legendre*[10] $(l_n)_{n\geq 0}$ est par définition la famille de fonctions polynomiales générée par le procédé d'orthogonalisation de Gram-Schmidt à partir du système libre $\{1, t, t^2, ..., t^k, ...\}$ des restrictions à $[-1,1]$ des fonctions monomiales, le produit scalaire étant

$$\langle f, g \rangle_{\text{leg}} = \int_{[-1,1]} f(t) \, \overline{g(t)} \, dt,$$

10. Ainsi nommés en l'honneur d'Adrien-Marie Legendre (1752-1833), mathématicien français qui les introduisit dans ses travaux sur les équations algébriques.

et normalisées suivant la convention

$$l_n(X) = \frac{(2n)!}{2^n (n!)^2} X^n + \cdots .$$

On vérifie par récurrence que

$$l_n(X) = \frac{1}{2^n n!} \Big(\frac{d}{dX} \Big)^n [(X^2 - 1)^n].$$

Pour construire à partir de la famille $(l_n)_{n\geq 0}$ un système orthonormé $(L_n)_{n\geq 0}$ (par rapport à l'orthogonalité induite par le choix du poids $\omega \equiv 1$ sur $[-1, 1]$), il convient de multiplier l_n par $\sqrt{n + 1/2}$ (donc $L_n(X) = \sqrt{n + 1/2} \times l_n(X)$).

On vérifie que les polynômes de Legendre vérifient une équation différentielle [11] du second ordre :

$$(t^2 - 1)\, l_n''(t) + 2t\, l_n'(t) - n(n + 1)\, l_n(t) \equiv 0, \;\; n = 0, 1, ... \tag{4.16}$$

On les calcule de proche en proche par la relation de rćurrence

$$(n + 1) l_{n+1}(X) = (2n + 1)\, X\, l_n(X) - n l_{n-1}(X). \tag{4.17}$$

La représentation graphique des polynômes de Legendre fait apparaitre une amplification de l'amplitude des oscillations au fur et à mesure que l'on se rapproche des bornes de l'intervalle $[-1, 1]$ (la fréquence augmentant elle aussi) avec de plus un maximum d'amplitude de plus en plus élevé au fur et à mesure que n augmente. Ceci tient au fait que le poids reste constant et ne permette donc pas de faire en sorte que puissent être atténués les *effets de bord*. Ceci constitue un point négatif concernant leur utilisation en algorithmique numérique (en particulier face au problème de l'approximation polynomiale uniforme). Voici par exemple (figure 1) le graphe d'un polynôme de Legendre (ici normalisé pour être d'énergie égale à 1) ; ici $L_{30}(-1) = L_{30}(1) = 5.5227$, $L_{30}(-.99) = L_{30}(.99) = -1.9805$, ce qui conforte notre constat.

11. On verra plus loin (dans le chapitre consacré aux équations différentielles) comment certaines équations du second ordre sur $\mathbb{R}$ génèrent des familles de polynômes orthogonaux, chaque élément de la famille correspondant à une valeur propre d'un opérateur différentiel du second ordre.

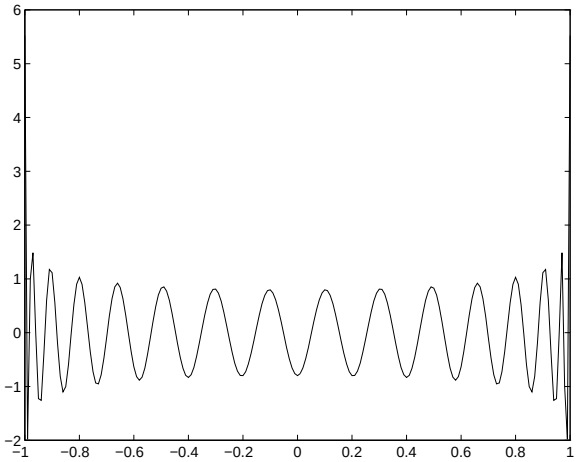

FIGURE 1. $s = L_{30}(t)$, $t = -1 : .01 : 1$

Un rôle important joué par les polynômes de Legendre concerne l'intégration en coordonnées sphériques dans $\mathbb{R}^3$ et la manière dont ces polynômes interviennent dans la représentation des *harmoniques sphériques*. Nous y reviendrons dans le prochain chapitre (Intégration et algorithmique numérique).

b) *Les polynômes de Tchebychev*

Ils sont déjà apparu dans ce cours, à l'occasion du rôle très important qu'ils jouent concernant le problème de l'approximation uniforme sur un segment (Section 4.2.3). On prend toujours $(a, b) = [-1, 1]$, mais le poids est maintenant

$$\omega(t) := \frac{1}{\sqrt{1-t^2}}$$

(notons qu'il tend vers $+\infty$ lorsque l'on s'approche des bones du segment).

DÉFINITION 4.12. La famille des *polynômes de Tchebychev* $(T_n)_{n\geq 0}$ est par définition la famille de fonctions polynomiales générée par le procédé d'orthogonalisation de Gram-Schmidt à partir du système libre $\{1, t, t^2, ..., t^k, ...\}$ des restrictions à $[-1, 1]$ des fonctions monomiales, le produit scalaire étant

$$\langle f, g\rangle_{\text{tch}} = \int_{[-1,1]} \frac{f(t)\overline{g(t)}}{\sqrt{1-t^2}}\, dt,$$

et normalisées suivant la convention

$$T_n(X) = 2^{n-1} X^n + \cdots,\ n \geq 1, \quad T_0(X) \equiv 1.$$

On vérifie immédiatement que T_n est le polynôme de degré n impliqué dans la relation trigonométrique

$$T_n(\cos\theta) = \cos(n\theta),$$

ce qui donne un moyen très simple d'en trouver les zéros : les zéros de T_n sont les images par cos des réels $\theta \in [0, 2\pi[$ tels que $\cos(n\theta) = 0$, *i.e* les points

$$\cos\frac{(2k-1)\pi}{2n}, \quad k = 1, ..., n.$$

Cette fois, la représentation graphique de T_n fait apparaitre des oscillations dont la fréquence augmente lorsque l'on se rapproche des extrémités -1 et 1 sans que cette fois l'amplitude de ces oscillations n'en soit affectée (voir la figure 2).

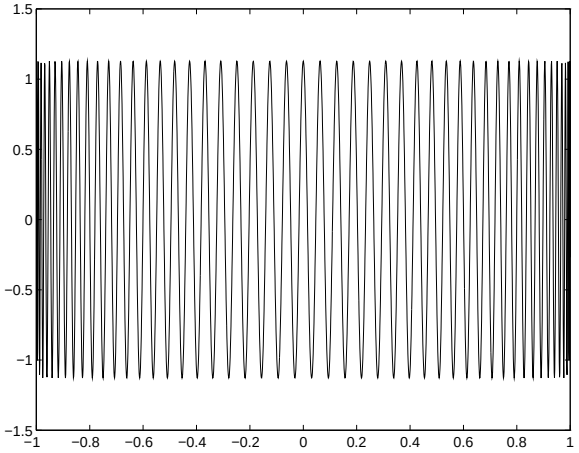

FIGURE 2. $s = T_{30}(t)$, $t = -1 : .01 : 1$

Outre leur rôle important en théorie de l'approximation uniforme (déjà vu dans la section 4.2.3), on retrouvera les polynômes de Tchebychev lorsqu'il s'agira de modéliser des boites noires ayant vocation à couper le plus efficacement (i.e. le plus brutalement) possible les hautes fréquences en électronique ou en traitement de l'information.

L'équation du second ordre dont T_n est solution est

$$(1 - t^2)\, T_n''(t) - t\, T_n'(t) + n^2\, T_n(t) \equiv 0. \tag{4.18}$$

Le calcul inductif des T_n se fait suivant la formule de récurrence

$$T_{n+1}(X) = 2\, X\, T_n(X) - T_{n-1}(X). \tag{4.19}$$

c) *Les polynômes de Hermite*

Les polynômes de Hermite [12] sont intrinsèquement liés à la transformation de Fourier (outil majeur en électronique, en optique ou en théorie de l'information, correspondant dans un cadre physique à la matérialisation de la *dualité*) et à la *fonction de Gauss*

$$t \mapsto \frac{1}{\sqrt{2\pi}}\, e^{-t^2/2},$$

densité de la loi normale réduite centrée, mais aussi fonction propre (de valeur propre correspondante $\sqrt{2\pi}$) de cette même transformation de Fourier de $L^2_{\mathbb{C}}(\mathbb{R})$ ($\mathbb{C}$-espace des signaux réels d'énergie finie) dans lui-même.

On prend comme intervalle $(a, b) = \mathbb{R}$ et comme poids ω la fonction $t \mapsto e^{-t^2}$.

12. Ils portent le nom du mathématicien français Charles Hermite (1822-1901), mais furent introduits auparavant par Lagrange.

DÉFINITION 4.13. La famille des *polynômes de Hermite* $(H_n)_{n\geq 0}$ est par définition la famille de fonctions polynomiales unitaires générée par le procédé d'orthogonalisation de Gram-Schmidt à partir du système libre $\{1, t, t^2, ..., t^k, ...\}$ des fonctions monomiales sur $\mathbb{R}$, le produit scalaire étant

$$\langle f, g\rangle_{\text{herm}} = \int_{\mathbb{R}} f(t)\overline{g(t)}\, e^{-t^2}\, dt.$$

On vérifie que

$$H_n(X) = \frac{(-1)^n}{2^n}\, e^{X^2} \times \Big(\frac{d}{dX}\Big)^n [e^{-X^2}],\ n = 0, 1, ..., \tag{4.20}$$

L'équation du second ordre auquel H_n satisfait est

$$H_n''(t) - 2t\, H_n'(t) + 2n\, H_n(t) \equiv 0. \tag{4.21}$$

La relation de récurrence permettant de calculer inductivement les polynômes H_n est

$$H_{n+1}(X) = X\, H_n(X) - \frac{n}{2}\, H_{n-1}(X). \tag{4.22}$$

Les polynômes de Hermite sont liées aux dérivées succcessives de la fonction de Gauss g. Cette fonction joue un rôle important en modélisation car, pour $\epsilon > 0$ petit ($\epsilon << 1$), la fonction

$$g_\epsilon\ : t \mapsto \frac{1}{\epsilon} g(t/\epsilon)$$

est un bon candidat pour modéliser l'impulsion [13] en $t = 0$. Les dérivées de g_ϵ, lorsque $\epsilon << 1$, sont candidates à modéliser les dérivées successives d'une impulsion, phénomènes se présentant, on le constate, comme des phénomènes de plus en plus oscillants.

Les polynômes de Hermite sont aussi liés aux séries génératrices exponentielles (voir Chapitre 2, Définition 2.1) par le biais de la formule

$$\exp(tz - z^2/2) = \sum_{n=0}^{\infty} \frac{H_n(t)}{n!}\, z^n, \quad t \in \mathbb{R},\ z \in \mathbb{C}. \tag{4.23}$$

La *fonction de Hermite*

$$t \mapsto h_n(t) := \frac{2^{n/2}}{\sqrt{n!}\pi^{1/4}}\, H_n(t)\, e^{-t^2/2}$$

est un vecteur propre unitaire ($\int_{\mathbb{R}} |h_n(t)|^2\, dt = 1$) de la transformation de Fourier qui à $f \in L^2_{\mathbb{C}}(\mathbb{R}, dt)$ associe la limite dans $L^2_{\mathbb{C}}(\mathbb{R}, d\omega)$ des fonctions

$$\omega \mapsto \int_{-N}^{N} f(t)\, e^{-i\omega t}\, dt$$

lorsque N tend vers $+\infty$, ce correspondant à la valeur propre $\sqrt{2\pi} \times (-i)^n$. Les fonctions de Hermite (et donc les polynômes de Hermite) jouent un rôle dans la réalisation de *transformation de Fourier fractionnaire* (on remplace $\sqrt{2\pi}(-i)^n = \sqrt{2\pi}\, e^{-in\pi/2}$ par $\sqrt{2\pi}\, e^{-i\alpha n\pi/2}$, $\alpha \in]0, 1[$). Pareille transformation est, comme la

13. Et par la même les pixels en traitement d'image ; la ≪ dérivation ≫ des pixels est une opération importante du point de vue informatique. Les fonctions oscillantes que sont les polynômes ou les fonctions de Hermite rendent compte de cette opération de dérivation.

dérivation fractionnaire que nous verrons plus loin, fréquemment utilisée en ingénierie ou en physique.

d) *Les polynômes de Laguerre*

Cette fois $(a,b) = [0,\infty[$ et $\omega(t) = e^{-t}$ (densité de la loi exponentielle).

DÉFINITION 4.14. La famille des *polynômes de Laguerre*[14] $(\mathcal{L}_n)_{n\geq 0}$ est par définition la famille de fonctions polynomiales orthonormale générée par le procédé d'orthogonalisation de Gram-Schmidt à partir du système libre $\{1, t, t^2, ..., t^k, ...\}$ des restrictions à $[0,+\infty[$ des fonctions monomiales, le produit scalaire étant cette fois

$$\langle f, g\rangle_{\rm lag} = \int_{[0,\infty[} f(t)\overline{g(t)}\, e^{-t}\, dt,$$

normalisées par le fait que le coefficient dominant vaut $(-1)^n/n!$.

On vérifie que

$$\mathcal{L}_n(X) = \frac{e^X}{n!}\Big(\frac{d}{dX}\Big)^n [X^n e^{-X}], \quad n = 0, 1, 2, ... \tag{4.24}$$

L'équation du second ordre satisfaite par $\mathcal{L}_n$ est

$$t\,\mathcal{L}_n''(t) + (1-t)\,\mathcal{L}_n'(t) - n\mathcal{L}_{n-1}(t) \equiv 0. \tag{4.25}$$

La formule inductive permettant de calculer de proche en proche les polynômes de Laguerre est

$$\mathcal{L}_{n+1}(X) = (2n+1-X)\,\mathcal{L}_n(X) - n^2\,\mathcal{L}_{n-1}(X). \tag{4.26}$$

Les polynômes de Laguerre sont aussi liés aux séries génératrices ordinaires (voir Chapitre 2, Définition 2.1) par le biais de la formule

$$\frac{1}{1-t}\exp\Big(-\frac{zt}{1-t}\Big) = \sum_{n=0}^{\infty} \mathcal{L}_n(t)\; z^n,\;\; t \in D(0,1),\;\; z \in \mathbb{C}. \tag{4.27}$$

Une autre forme ≪ close ≫ de $\mathcal{L}_n$ est donnée par

$$\mathcal{L}_n(X) = \sum_{k=0}^{n} (-1)^k \binom{n}{k} \frac{X^k}{k!}. \tag{4.28}$$

4.4. Autour de la méthode des ≪ moindres carrés ≫

La méthode des moindres carrés soutend une quantité de démarches en algorithmique numérique, toutes d'usage très fréquent aujourd'hui au carrefour des mathématiques et de l'informatique : meilleure approximation quadratique, algorithmes itératifs à base d'itération de projections orthogonales, pseudo inversion de systèmes sous-déterminés à base de *décomposition en valeurs singulières* [S.V.D][15], algorithmes du type ≪ gloutons ≫ du type *Matching Pursuit*, réseaux de neurones et *Décomposition en modes propres* [P.O.D][16], Analyse en Composantes Principales, algorithmes dits ≪ génétiques ≫, etc. Nous nous contenterons dans ce cours d'esquisser quelques idées clef.

14. Introduits par le mathématicien français Edmond Laguerre (1834–1886).
15. *Singular Value Decomposition.*
16. *Proper Orthogonal Decomposition.*

4.4.1. Meilleure approximation quadratique. Le modèle ultra classique (que vous connaissez sûrement tous depuis le lycée) est la recherche de la *droite de régression* d'un nuage de points. Ces points (des couples (x_k, y_k), $k = 0, ..., N$), représentent par exemple les réponses chiffrées à deux questions d'une même enquête. Pour voir quel est le degré de corrélation des deux listes de résultats, on cherche la *droite de régression* du nuage, c'est-à-dire le couple de nombre réels (α, β) tel que l'expression quadratique

$$\sum_{k=0}^{N} |y_k - \alpha x_k - \beta_k|^2$$

soit minimale (d'où la terminologie « moindres carrés »). La valeur du minimum de cette expression mesure la *dispersion* (les économistes parlent de *volatilité*) du nuage. Plus cette dispersion est petite, plus on pourra affirmer que les résultats à la question 2 sont corrélés de manière affine à ceux de la question 1. Si l'on pose

$$m_x = \frac{\sum_{k=0}^{N} x_k}{N+1}, \quad m_y = \frac{\sum_{k=0}^{N} y_k}{N+1}$$

(les moyennes) et

$$\sigma_x^2 := \sum_{k=0}^{N} (x_k - m_x)^2, \quad \sigma_y^2 := \sum_{k=0}^{N} (y_k - m_y)^2$$

(les variances), l'équation de la droite de régression $y = \alpha x + \beta$ s'exprime comme

$$y - m_y = \rho\,(x - m_x),$$

où

$$\rho := \frac{\sum_{k=0}^{N} (x_k - m_x)(y_k - m_y)}{\sigma_x\,\sigma_y}$$

est appelé *coefficient de corrélation* du nuage de points.

On peut chercher aussi la meilleure fonction polynomiale de degré n prescrit (lorsque $0 \leq n \leq N$)

$$x \mapsto \alpha_0 + \alpha_1 x + \cdots + \alpha_n x,$$

« meilleure » au sens suivant : l'erreur quadratique

$$\sum_{k=0}^{N} \Big(y_k - \sum_{j=0}^{n} \alpha_j x_k^j \Big)^2 \tag{4.29}$$

est minimale. Si l'on prend $n = N$ et que les x_k sont $N + 1$ points distincts, on trouve comme fonction polynomiale le polynôme de Lagrange et l'erreur quadratique minimale vaut 0. Mais souvent au prix d'un polynôme de degré trop grand, avec tous les problèmes numériques qui sont inhérents à cette difficulté ! C'est la routine

```
>> P = polyfit(x,y,n)
```

qui, sous `Matlab` ou `Scilab`, assure la recherche de cette meilleure fonction polynomiale.

On suppose toujours les x_k distincts. Trouver les coefficients α_j, $j = 0, ..., n$ de manière à rendre minimale l'expression quadratique (4.29) relève d'un calcul de projection orthogonale sur un sous-espace de $\mathbb{R}^{N+1}$ équipé d'un produit scalaire. L'espace $\mathbb{R}^{N+1}$ doit être ici compris comme le $\mathbb{R}$-espace vectoriel des fonctions à

valeurs réelles définies sur l'ensemble fini $\{x_0, ..., x_N\}$ (de cardinal $N+1$). Le vecteur $(\xi_0, ..., \xi_N)$ correspond à la fonction valant ξ_j en x_j, $j = 0, ..., N$. Dans ce $\mathbb{R}$-espace vectoriel $\mathbb{R}^{N+1}$, les restrictions à $\{x_0, ..., x_N\}$ des fonctions $x \mapsto x^j$, $j = 0, ..., n$, engendrent un sous-espace V_n de dimension $n + 1$. Si l'on prend comme produit scalaire sur $\mathbb{R}^{N+1}$ le produit scalaire usuel, on voit que la restriction à $\{x_0, ..., x_N\}$ de

$$x \mapsto \sum_{j=0}^{n} \alpha_j \, x^j$$

doit être, pour que l'expression (4.29) soit minimale, exactement la projection sur V_n de la fonction qui à x_k associe y_k, $k = 0, ..., N$. Le système linéaire à résoudre pour calcuuler les α_j est le système de Cramer (en les α_j)

$$\sum_{k=0}^{N} \Big(y_k - \sum_{j=0}^{n} \alpha_j x_k^j \Big) \, x_k^l = 0 \quad l = 0, ..., n \, .$$

On peut aussi remarquer que, dans le contexte d'un intervalle (a, b) et d'un poids $\omega \ : (a, b) \to [0, \infty]$ détaillé dans la section 4.3.1, la recherche, étant donné une fonction f continue sur (a, b), du ≪ meilleur ≫ polynôme p (parmi les polynômes de degré n prescrit), au sens où l'expression quadratique

$$\int_{(a,b)} |f(t) - p(t)|^2 \, \omega(t) \, dt$$

soit minimale, correspond aussi à un calcul de projection orthogonale, celui de f sur le sous-espace vectoriel de $C((a, b), \mathbb{R})$ engendré par les fonctions monômes $1, t, ..., t^n$, donc aussi par les polynômes orthogonaux $P_0, ..., P_n$ attachés aux donnés $((a, b), \omega)$ et construits comme dans la section 4.3.1. Comme on l'a vu, le polynôme optimal p cherché est donné par

$$p = \sum_{k=0}^{n} \frac{\int_{(a,b)} f(t) \, \overline{P_k(t)} \, \omega(t) \, dt}{\int_{(a,b)} |P_k(t)|^2 \, \omega(t) \, dt} \, P_k.$$

4.4.2. Une introduction aux algorithmes itératifs à base de projections orthogonales itérées. Dans cette section, nous étendons l'idée de projection orthogonale en lui adjoignant le concept d'algorithme itératif, pour présenter l'idée sous jacente à l'*algorithme de Kaczmarz*. Pareil algorithme a de nombreuses incarnations en algorithmique numérique, au carrefour des mathématiques et de l'informatique, et mérite que l'on s'y attarde un instant.

Le cadre est le suivant. On dispose de p opérateurs linéaires (supposés surjectifs)

$$R_j \ : \mathbb{R}^N \to \mathbb{R}^{M_j}, \ j = 1, ..., p.$$

On suppose $N >> M_j$. Les espaces $\mathbb{R}^N$ et $\mathbb{R}^{M_j}$ sont équipés du produit scalaire usuel. Une entrée $X \in \mathbb{R}^N$ est inconnue, mais on dispose de la connaissance des $R_j \cdot X$.

Exemple 4.15 (l'exemple du CAT-Scanner). Un exemple peut être fourni par le CAT-Scanner en instrumentation médicale. L'entrée X est la densité de rayonnement d'un organe $3D$ chargé. Les $R_j \cdot X$, $j = 1, ..., p$, sont les images $2D$ obtenues après positionnement du dispositif d'enregistrement du CAT-Scanner suivant un angle θ_j donné (en principe, il faut balayer tout $[0, 2\pi]$ pour espérer reconstituer

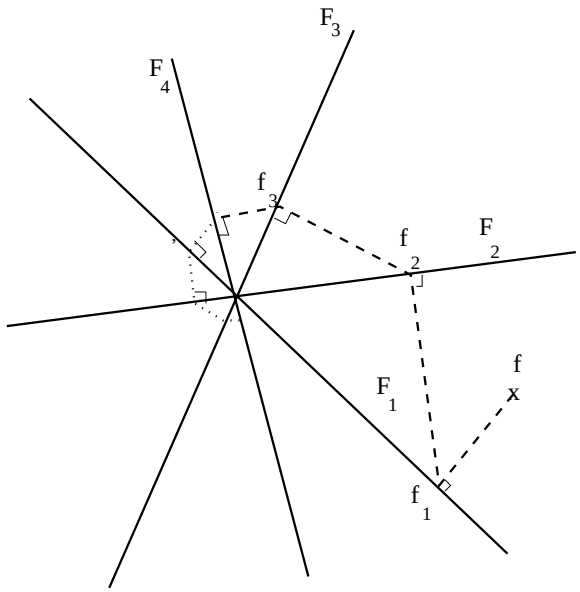

FIGURE 3. Une vision heuristique de la méthode de Kaczmarz

X à partir des $R_j \cdot X$). Il est clair que X obéit à des contraintes (avoir toutes ses coordonnées positives entre autres).

On note F_j le noyau de R_j, $j = 1, ..., p$. Quand bien même X est inconnu, la connaissance de $R_j \cdot X = g_j$, $j = 1, ..., p$, induit la possibilité de calculer, étant donné un vecteur f quelconque de $\mathbb{R}^N$, la projection orthogonale de f sur le sous-espace affine $X + F_j$. On a en effet

$$f - \mathrm{Pr}_{X+F_j}[f] \in F_j^{\perp} = \mathrm{Im}\, R_j^*,$$

donc

$$f - \mathrm{Pr}_{X+F_j}[f] = R_j^*[u]. \tag{4.30}$$

L'opérateur $R_j \circ R_j^* : \mathbb{R}^{M_j} \to \mathbb{R}^{M_j}$ est inversible (car injectif, puisque l'on a supposé R_j surjectif) et u se calcule donc en appliquant R_j aux deux membres de (4.30), ce qui donne

$$R_j \cdot f - R_j \cdot \mathrm{Pr}_{X+F_j}[f] = R_j \cdot f - g_j = (R_j \circ R_j^*)[u],$$

puis en appliquant aux deux membres de cette dernière formule l'inverse de $R_j \circ R_j^*$. On reporte enfin dans (4.30) pour trouver $\mathrm{Pr}_{X+F_j}[f]$. Pour simplifier, on note Q_j l'opérateur de projection orthogonale sur $X + F_j$.
La propriété (assez facile à établir et dont on pourra se convaincre graphiquement en examinant la figure 3) suivant laquelle la suite d'opérateurs $(Q_p \circ \cdots \circ Q_1)^k$, $k = 1, 2, ...,$ converge fortement vers la projection orthogonale sur l'intersection des sous-espaces $X + F_j = \mathrm{Ker}\,(\mathrm{Id}_{\mathbb{R}^N} - Q_j)$, $j = 1, ..., p$, c'est-à-dire que pour tout f dans $\mathbb{R}^N$,

$$\lim_{k \to +\infty} (Q_p \circ \cdots \circ Q_1)^k \cdot f = \mathrm{Pr}_{X+F_1 \cap \cdots \cap F_p}[f]\,. \tag{4.31}$$

Cette propriété subsiste lorsque l'on introduit le paramètre de relaxation et que l'on remplace chaque Q_j par

$$Q_{j,\varpi} = (1-\varpi)\mathrm{Id}_{\mathbb{R}^N} + \varpi\, Q_j.$$

On note cependant que

$$\|Q_j^{\varpi}\cdot f\|^2 - \|f\|^2 = (2-\varpi)\varpi\,(\|Q_j\cdot f\|^2 - \|f\|^2) \quad \forall\, f\in\mathbb{R}^N \tag{4.32}$$

et que l'on doit se restreindre à $\varpi \in]0,2[$ pour le choix de ce paramètre de relaxation. Si l'on choisit d'initier l'algorithme à partir de $f=0$ (ou plus généralement de $f\in\bigoplus_{j=1}^p F_j^{\perp}$), alors on atteint ainsi asymptotiquement la solution $\widetilde{X}$ du système $R_j\cdot\widetilde{X} = g_j$, $j=1,...,p$, qui est de norme quadratique minimale.

Il faut noter enfin que cette méthode itérative permet, à chaque itération, de préciser des contraintes sur X en les faisant porter sur la solution approchante trouvée. En bref, ce type de démarche s'accompagne d'une grande souplesse. Le défaut est le risque d'instabilité numérique croissante (tenant compte de bruit accompagnant l'enregistrement des $R_j\cdot X$) au fur et à mesure que le nombre d'itérations augmente. L'algorithme doit être arrêté à partir de ce moment.

4.4.3. La décomposition en valeurs singulières et son efficacité. La *décomposition en valeurs singulières* (plus communément ≪ SVD ≫ pour *Singular Value Decomposition*) s'avère être un outil efficace d'algèbre linéaire, en prise, comme la méthode de Kaczmarz présentée dans la sous-section précédente, avec la minimisation quadratique, outil que l'on exploite souvent au carrefour des mathématiques et de l'informatique. Nous en présentons ici une brève introduction.

Soit R une application linéaire (supposée ici surjective pour simplifier, comme les R_j dans la sous-section précédente) de $\mathbb{R}^N$ dans $\mathbb{R}^M$, représentée par une matrice A lorsque $\mathbb{R}^N$ et $\mathbb{R}^M$ sont rapportés à leurs bases canoniques respectives (l'hypothèse de surjectivité implique évidemment $N\geq M$). D'après la formule du rang, le noyau de R est un sous-espace vectoriel F de $\mathbb{R}^N$ de dimension $N-M$, que l'on peut donc rapporter à une base orthonormée $(e_{M+1},...,e_N)$ (pour le produit scalaire canonique). Si l'on complète (suivant le procédé de Gram-Schmidt) cette base en une base orthonormée $(e_1,...,e_N)$ de $\mathbb{R}^N$, on constate que $(e_1,...,e_M)$ constitue une base du sous-espace du sous-espace $F^{\perp}$ constitué des vecteurs de $\mathbb{R}^N$ orthogonaux au noyau de R. La restriction de R à $F^{\perp}$ est une application linéaire bijective entre $F^{\perp}$ et $\mathbb{R}^M$ (en effet R est ici supposée surjective). Si $\mathbb{R}^M$ est rapporté à la base $(e_1,...,e_M)$ et $\mathbb{R}^N$ à sa base canonique, la matrice de R dans ces bases s'écrit sous la forme :

$$\begin{pmatrix} & 0 & \cdots & \cdots & 0\\ B & \vdots & \vdots & \vdots & \vdots\\ & 0 & \cdots & \cdots & 0\end{pmatrix}$$

où B est une matrice réelle $M\times M$ de déterminant non nul ; il existe donc une matrice orthogonale réelle V_1 de taille (N,N) telle que

$$A = \begin{pmatrix} & 0 & \cdots & \cdots & 0\\ B & \vdots & \vdots & \vdots & \vdots\\ & 0 & \cdots & \cdots & 0\end{pmatrix}\cdot V_1^*$$

(on note C^* la trans-conjuguée d'une matrice C, c'est-à-dire la transposée ${}^tC$ de C si C, comme ici V_1, est réelle). La matrice $B \cdot B^*$ (ici encore $B^* = {}^tB$ car B est réelle) est une matrice symétrique réelle que l'on peut donc écrire

$$B \cdot B^* = U \cdot \mathrm{diag}\,(\lambda_1, ..., \lambda_p) \cdot U^*\,,$$

où U est une matrice orthogonale réelle de taille (M, M) ; d'autre part, pour tout $v \in \mathbb{R}^M$,

$$\langle B \cdot B^*(v)\,,\, v\rangle = \langle B^*(v)\,,\, B^*(v)\rangle \geq 0\,,$$

ce qui implique que pour $j = 1, ..., M$, $\lambda_j = \sigma_j^2 \geq 0$ (on appelle σ_j la racine positive du réel positif λ_j) ; on peut d'ailleurs faire en sorte que $\sigma_1^2 \geq \sigma_2^2 \geq \cdots \geq \sigma_M^2$, ce que l'on fera. Comme $\det B \neq 0$, ces nombres réels positifs σ_j^2, $j = 1, ..., M$ sont tous non nuls. Les nombres réels positifs $\sigma_1, ..., \sigma_M$ correspondants sont dits *valeurs singulières* de l'application linéaire R (il est clair que ces nombres ne dépendent que de R et non du choix des bases dans lesquelles l'action de R est exprimée matriciellement). Comme

$$B \cdot B^* = \Big(U \cdot \mathrm{diag}\,(\sigma_1, ..., \sigma_M)\Big) \cdot \Big(U \bullet \mathrm{diag}\,(\sigma_1, ..., \sigma_M)\Big)^*\,,$$

la matrice

$$B^* \cdot \left(\Big(U \cdot \mathrm{diag}\,(\sigma_1, ..., \sigma_M)\Big)^*\right)^{-1}$$

est une matrice orthogonale réelle W de taille (M, M) et l'on peut donc écrire

$$B^* = W \cdot \mathrm{diag}\,(\sigma_1, ..., \sigma_M) \cdot U^*\,,$$

soit

$$B = U \cdot \mathrm{diag}\,(\sigma_1, ..., \sigma_M) \cdot W^*.$$

On peut donc ainsi écrire

$$A = U \cdot \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \sigma_M & 0 & \cdots & \cdots & 0 \end{pmatrix} \cdot V^* \tag{4.33}$$

où U et V sont respectivement des matrices orthogonales réelles de tailles (M, M) et (N, N). C'est cette représentation (4.33) que l'on qualifie de *décomposition en valeurs singulières* de la matrice A, matrice de l'opérateur linéaire R. Sous un environnement de calcul scientifique (comme `MATLAB` ou `Scilab`), on la réalise à partir de la matrice d'entrée A sous la commande :

```
>>[U,D,V] = svd(A);
```

La matrice D fournit les valeurs singulières, les M vecteurs colonne de U constituent le système orthonormé (de l'espace but $\mathbb{R}^M$) choisi pour exprimer sous la forme (4.33) l'action de l'opérateur linéaire (de matrice A dans les bases canoniques), tandis que les N vecteurs colonnes de V constituent le système orthonormé (de l'espace source $\mathbb{R}^N$ cette fois) qui lui est couplé. Cette transformation permet (quitte à effectuer des transformations orthogonales à la source et au but) de ramener l'étude d'une application linéaire surjective de $\mathbb{R}^N$ dans $\mathbb{R}^M$ à celle d'une application linéaire dont la matrice est une matrice diagonale inversible de taille (M, M), complétée à droite par une matrice de zéros de taille $(M, N - M)$. C'est donc un outil très intéressant du point de vue pratique (parfois trop méconnu). On se doute en particulier du rôle important joué par les vecteurs colonnes de V,

vecteurs ≪ modèles ≫ de l'espace source $\mathbb{R}^N$ lorsqu'il est soumis à la transformation linéaire R de matrice A.

Si R n'est pas surjective, cette décomposition (4.33) reste valide (mais $M - \text{rang}\,(R)$ valeurs singulières σ_j sont alors nulles, à savoir les $M - \text{rang}\, R$ dernières).

Tout ceci peut se transposer au cadre complexe (R tant une application $\mathbb{C}$-linéaire de $\mathbb{C}^N$ dans $\mathbb{C}^M$) ; il suffit juste de remplacer ≪ orthogonale ≫ par ≪ unitaire ≫. Les valeurs singulières restent des nombres strictement positifs.

Les valeurs singulières d'une matrice carrée B réelle ou complexe (supposée ici inversible) de taille (M, M) jouent aussi un rôle important pour exprimer le *conditionnement* de cette matrice, relativement au choix de la norme euclidienne

$$\|y\|_2 = \sqrt{y_1^1 + \cdots + y_M^2}$$

dans $\mathbb{R}^M$ (cette norme est la norme par défaut `norm` dans la plupart des logiciels de calcul scientifique, tels `MATLAB` ou `Scilab`). Le choix de cette norme induit le choix d'une norme sur les matrices carrées de taille (n, n) par

$$\|C\|_2 := \sup_{\{y \in \mathbb{R}^M \,;\, \|y\|_2 = 1\}} \|C \cdot y\|_2$$

et le conditionnement d'une matrice (M, M) inversible B (relativement à ce choix de norme euclidienne sur $\mathbb{R}^M$) est par définition le nombre

$$\text{cond}_{\|\ \|_2}\, B := \|B\| \times \|B^{-1}\|.$$

L'instabilité numérique que l'on rencontre dans la résolution de systèmes linéaires du type $B \cdot Y = Z$ (voir [**Y1**], Sections 1.6 et 4.5) est liée au fait que ce conditionnement est grand. Plus c'est le cas, moins la résolution numérique de $B \cdot Y = Z$ est stable, donc fiable (on dit dans ce cas que B est *mal conditionnée*). Le conditionnement d'une telle matrice (M, M) inversible B est précisément donné par

$$\text{cond}_{\|\ \|_2}\, B = \frac{\sigma_1(B)}{\sigma_M(B)},$$

où $\sigma_1(B) \geq \cdots \geq \sigma_M(B) > 0$ désigne la liste des valeurs singulières de B rangées dans l'ordre décroissant.

Revenons pour finir au cas général d'un opérateur linéaire surjectif R de $\mathbb{R}^N$ dans $\mathbb{R}^M$. Voici comment on exploite la décomposition en valeurs singulières (4.33) de R. Notons $u_1, ..., u_M$ les vecteurs colonne de U, $v_1, ..., v_N$ les vecteurs colonne de V. Si Y est un vecteur de $\mathbb{R}^M$ donné, on remarque que

$$X = \sum_{k=1}^{M} \frac{1}{\sigma_k} \langle Y, u_k \rangle\, v_k \tag{4.34}$$

est le vecteur de norme (euclidienne) minimale parmi tous les vecteurs f minimisant

$$f \in \mathbb{R}^N \longmapsto \|R[f] - Y\|^2,$$

donc un candidat envisageable comme antécédent de Y *via* l'application R. Cet élément (4.34) est appelé *pseudo-inverse* de Y pour l'opérateur linéaire R. Si R

n'est plus surjective, ceci vaut encore, mais l'on doit alors choisir de définir le pseudo-inverse X par

$$X = \sum_{k=1}^{\text{rang}(R)} \frac{1}{\sigma_k} \langle Y, u_k \rangle \, v_k \tag{4.35}$$

4.4.4. Les algorithmes « gloutons » et la minimisation quadratique. Les algorithmes dits « gloutons » sont de plus en plus utilisés dans des questions à l'interface des mathématiques et de l'informatique : traque d'une image médicale contre un « dictionnaire » d'images pathologiques, poursuite d'un visage ou d'une scène dans un flux vidéo, réseaux de neurones, algorithmes dits « génétiques », *etc.* On en donne ici une rapide présentation, qui sera plutôt développée plus avant au titre de projet.

Si l'orthogonalité est une notion clef dans tout processus d'analyse et de synthèse en théorie de l'information, c'est malheureusement une notion « fragile » : par exemple le procédé d'orthonormalisation de Gram-Schmidt dans $\mathbb{R}^N$, initié à partir d'un système libre au sein duquel duquel deux vecteurs sont « presque » colinéaires, peut générer des combinaisons de ces vecteurs de base affectés d'énormes coefficients ! Pour prendre un exemple plus concret, il est souvent utile aujourd'hui, face aux questions de sécurité ou de *copyright*, d'authentifier une image en y sur-imprimant un code (une « marque »[17]) que seul son propriétaire est à même de déceler ; si la réalisation préalable de décompositions orthogonales permet de « signer » l'image en codant intelligemment les divers composants de l'une de ces décompositions, il n'est aucunement évident que l'orthogonalité entre composants soit un tant soit peu préservée lorsque l'image subit un quelconque traitement (compression, modification géométrique, filtrage, *etc.*) ; dès lors, le processus d'authentification de l'image perturbée par son propriétaire devient impossible ! Voici encore un second exemple : en imagerie médicale, il s'avère fréquemment utile, plutôt que de tenter d'inverser le mécanisme tomographique $\mathcal{R}$ qui a permis d'obtenir une image I, de chercher à « pister » cette image contre un dictionnaire (éventuellement redondant, mais on veillera à limiter cela) d'images $\mathcal{R}[f_j]$, où $f_1, ..., f_N$ sont des états source pathologiques auxquels on aimerait confronter l'état f dont on ne connait que $\mathcal{R}[f]$; on pressent en effet que f s'apparente à une combinaison de ces états pathologiques f_j, $j = 1, ..., N$, combinaison qu'il serait judicieux de retrouver sous forme « organisée » : d'abord la pathologie (parmi les f_j) la mieux « corrélée » à f, puis celle qui, bien qu'aussi corrélée signicativement à f, l'est un peu moins, *etc.* Ce sont ces idées que nous allons esquisser dans cette section.

Étant donnée, dans $\mathbb{R}^N$, un vecteur X dont on souhaite extraire une information (aux fins d'analyse, de classification, ou de synthèse ultérieure), l'un des algorithmes les plus naifs (mais de fait aussi les plus robustes) que l'on puisse imaginer est celui qui consiste à « pister » X contre un « dictionnaire » que l'on aura au préalable composé d'éléments « test » dont on s'attend *a priori* que l'élément X proposé soit une combinaison linéaire. Il peut s'avérer utile (on le verra plus loin) d'élaborer un dictionnaire d'éléments « test » à partir de l'élément X que l'on prétend analyser. On donnera dans cette section une version élémentaire de

17. C'est ce que l'on appelle le « *watermarking* ».

l'algorithme mathématique soutendant ce scénario, dit algorithme de ≪ *Matching Pursuit* ≫ [18] .

On convient d'appeler *dictionnaire* une liste finie d'éléments de $\mathbb{R}^N$ (certainement assez redondante) $\{d_1, ..., d_L\}$; pour simplifier, on suppose tous les éléments du dictionnaire normalisés et de norme euclidienne égale à 1. Étant donné un tel dictionnaire $\mathcal{D}$ et un élément X de $\mathbb{R}^N$, l'algorithme le plus simple proposé (MP pour ≪ *Matching Pursuit*) ≫ est décrit par le scénario suivant :

(1) On commence par calculer en famille toutes les corrélations $\langle X, d_j\rangle$ pour $j = 1, ..., L$ et l'on choisit, parmi tous les éléments-test d_j, un élément d tel que
$$|\langle X, d\rangle| = \min_{j=1,...,L} |\langle X, d_j\rangle|.$$
Pareil élément d constitue, au sein du dictionnaire, l'un de ceux (il peut y en avoir plusieurs) les mieux corrélés au vecteur X. On décide d'appeler cet élément d_1, donc de renuméroter le dictionnaire.

(2) On calcule dans un second temps le vecteur
$$\texttt{Reste}\,[1] = X - \langle X, d_1\rangle\, d_1 = X - \mathrm{Proj}_{\mathbb{R}d_1} X$$
et
$$\texttt{Resume}\,[1] = \langle X, d_1\rangle\, d_1 = \mathrm{Proj}_{\mathbb{R}d_1} X.$$

(3) On calcule toutes les corrélations $\langle \texttt{Reste}\,[1], d_j\rangle$, $j = 1, ..., L$, afin de détecter un élément-test d (on l'appellera d_2) tel que
$$|\langle \texttt{Reste}\,[1], d\rangle| = \min_{j=1,...,L} |\langle \texttt{Reste}\,[1], d_j\rangle|$$
(rien n'interdit *a priori* que l'on retrouve d_1).

(4) On pose alors
$$\texttt{Reste}\,[2] = \texttt{Reste}\,[1] - \langle \texttt{Reste}\,[1], d_2\rangle\, d_2$$
et
$$\texttt{Resume}\,[2] = \texttt{Resume}\,[1] + \langle \texttt{Reste}\,[1], d_2\rangle\, d_2.$$

(5) On recommence l'opération avec $\texttt{Reste}\,[2]$ pour trouver d_3, etc.

En calculant de proche en proche les résumés successifs $\texttt{Resume}\,[k]$, $k = 1, 2, ...$, on voit se ≪ recomposer ≫ X suivant présicément de dictionnaire $\mathcal{D}$, les éléments test étant pris dans l'ordre de leurs corrélations décroissantes avec l'information X à tester.

Pareil algorithme peut être significativement amélioré. Il est en effet entaché d'un défaut : il faut à chaque étape travailler avec le dictionnaire complet et l'on ne peut se permettre d'éliminer du dictionnaire les éléments test dès lors qu'ils sont apparus ; pour corriger cet état de fait et améliorer (attention, pas toujours cependant !) l'algorithme de *Matching Pursuit*, on peut en introduire une version orthogonale en le couplant avec le procédé d'orthonormalisation de Gram-Schmidt : c'est l'algorithme *Matching Pursuit Orthogonal* (MPO). Il s'agit, outre la démarche décrite plus haut, d'imposer à chaque itération (à savoir la détection de d_{k+1}) l'orthogonalité du reste avec les éléments du dictionnaire précédemment sélectionnés ; pareil

18. La terminologie anglo-saxonne résume l'objectif : ≪ traquer ≫ (ou ≪ poursuivre ≫) en essayant d'≪ ajuster ≫ aux combinaisons déléments du dictionnaire (c'est la phase de ≪ *matching* ≫).

algorithme nécessite bien sûr, dès cet élément test d_{k+1} déterminé, un ré-ajustement des coefficients du « résumé » obtenu précédemment. Son implémentation est proposée au titre du projet.

Il peut s'avérer intelligent, plutôt que de confronter l'information X à étudier contre un dictionnaire *a priori*, de chercher à construire un dictionnaire *raisonné* à partir de l'information X elle même, en ce basant sur une démarche d'inspiration statistique. C'est l'idée de la *Décomposition en Modes Propres* (« *Proper Orthogonal Decomposition* », en abrégé POD), que l'on retrouve aussi dans les démarches algorithmiques relevant des réseaux de neurones. Ici encore, la construction d'un tel dictionnaire raisonné fera l'objet d'un sujet de projet. Il se trouve que cette construction (rejoignant ce que l'on appelle dans le jargon statistique l'*Analyse en Composantes Principales*) est directement reliée à la décomposition en valeurs singulières présentée dans la sous-section précédente.

4.5. L'interpolation par des polynômes trigonométriques

Comme on l'a vu dans la sous-section 4.2.3, la recherche de la meilleure approximation polynomiale de t^{n+1} par des fonctions polynomiales de degré au plus n fait apparaitre des phénomènes oscillants (théorème 4.9 d'alternance de Tchebychev). Les fonctions oscillantes, du type

$$t \longmapsto \sum_{k=1}^{M} a_k\, e^{i\omega_k t},\ \omega_k \in \mathbb{R},\ a_k = \alpha_k e^{i\varphi_k},\ \alpha_k > 0\,,\ \varphi_k \in [0, 2\pi[\tag{4.36}$$

(cadre complexe) ou

$$t \longmapsto \sum_{k=1}^{M} \alpha_k\, \cos(\omega_k t + \varphi_k),\ \omega_k \in \mathbb{R},\ \alpha_k > 0\,,\ \varphi_k \in [0, 2\pi[\tag{4.37}$$

(cadre réel), où les α_k sont qualifiées d'*amplitudes*, les ω_k de *fréquences*, les φ_k de *phases*, jouent en théorie de l'information ou en physique (électronique, télécommunications, acoustique, mécanique ondulatoire, électromagnétisme, *etc.*) un rôle bien plus important que celui joué par les fonctions polynomiales. On peut donc se poser naturellement le problème de l'interpolation (aux fins de modélisation, d'analyse ou de synthèse, de compression, ...) de valeurs numériques $y_0, ..., y_N$ prises au dessus d'un maillage $x_0 < x_1 < \cdots < x_N$ par un polynôme trigonométrique à ensemble de fréquences $\{\omega_1, ..., \omega_M\}$ prescrit.

4.5.1. le problème du sous-échantillonnage. Un problème sérieux apparait à ce propos : si le pas du maillage est trop grand, les valeurs aux points de ce maillage d'un polynôme trigonométrique P impliquant de trop grandes fréquences (*i.e.* oscillant trop vite) ne sauraient suffire à rendre compte du polynôme ! C'est le délicat problème du *sous-échantillonnage* (voir la figure 4).

Pour énoncer cependant un résultat positif rapport à l'écueil que représente ce problème, il nous faut brièvement rappeler ce qu'est la transformée de Fourier (ou encore le *spectre*) d'une information $f\ : \mathbb{R} \mapsto \mathbb{C}$ que l'on suppose nulle hors de $[-N\tau, N\tau]$ pour $N \in \mathbb{N}$ assez grand, τ représentant un pas (fixe) de maillage, et de module intégrable sur $[-N\tau, N\tau]$. Par définition, ce spectre est la fonction

$$\omega \mapsto \int_{[-N\tau, N\tau]} f(t)\, e^{-i\omega t}\, dt = \langle f, e^{-i\omega t} \rangle.$$

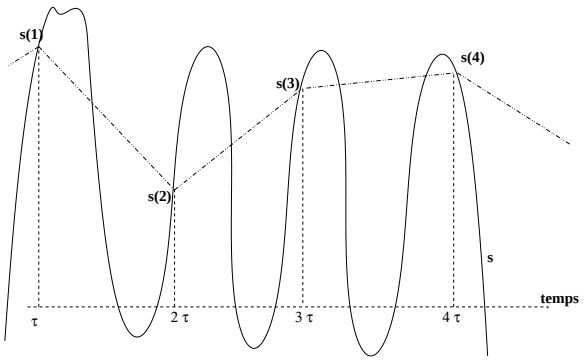

FIGURE 4. Face à un problème : le sous-échantillonnage

Dire que $|\widehat{f}(\omega)|$ est petit signifie que phénomène oscillant élémentaire $t \mapsto e^{i\omega t}$ est peu corrélé à l'information f. Dire au contraire que $|\widehat{f}(\omega)|$ est grand signifie que f et ce phénomène oscillant sont fortement correlées, ou encore que la fréquence ω est significativement présente dans l'analyse spectrale de l'information f, ou bien encore que $t \mapsto e^{i\omega t}$ est une composante harmonique significative dans la décomposition de l'information f « en harmoniques » (penser à l'acoustique par exemple, au cas où f est le rendu sonore d'un enregistrement orchestral). Nous avons le résultat mathématique suivant (que nous admettrons).

THEORÈME 4.16 (théorème d'échantillonnage de Shannon-Nyquist). *S'il existe des fréquences ω telles que $|\widehat{f}(\omega)| > 0$ et $|\omega| > \Omega$, l'information f est sous-échantillonnée aux points $k\tau$, $k = -N, ..., N$, dès que $\tau > \pi/\Omega$. Cependant, si*

$$\int_{|\omega|>\Omega} |\widehat{f}(\omega)|\, d\omega < \pi\epsilon$$

avec $\epsilon \geq 0$ et si $\tau \leq \pi/\Omega$, la restitution approchée de l'information f (à une erreur uniforme d'au plus ϵ) à partir de ses échantillons $f(k\tau)$, $k = -N, ..., N$, est possible et donnée par

$$f(t) \simeq \sum_{k=-N}^{N} f(k\tau)\, \mathrm{sinc}\Big(\frac{t - k\tau}{\tau}\Big), \tag{4.38}$$

où la fonction sinuscardinal sinc *est*[19] *la fonction*

$$u \in \mathbb{R} \longmapsto \frac{\sin \pi u}{\pi u}.$$

REMARQUE 4.17. La formule (4.38) devient exacte si f n'a pas de composantes fréquentielles (positives ou négatives) au dessus du seuil Ω et si $\tau \leq \pi/\Omega$. On note toutefois que ceci relève de l'utopie car tout phénomène physique est en général entaché de bruit (erreur de mesure lors de l'enregistrement, bruit inhérent au phénomène, *etc.*) et que le bruit fait apparaitre des composantes fréquencielles de fréquences arbitrairement grandes. Le seuil maximal toléré pour τ, c'est-à-dire π/Ω est alors appelé *seuil de Nyquist*.

19. On convient d'adopter la normalisation utilisée sous `MATLAB`.

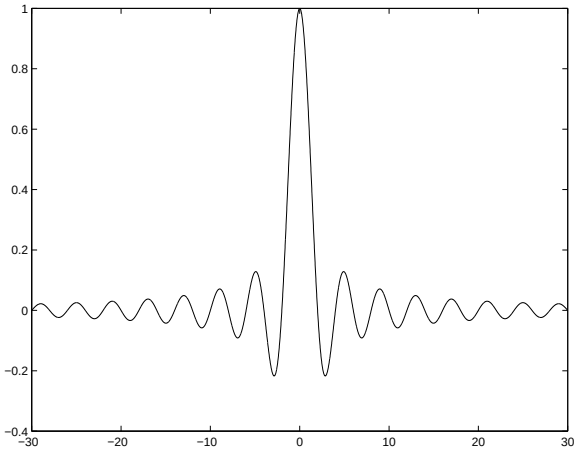

FIGURE 5. Le graphe de $u \mapsto \operatorname{sinc} u$

Le graphe de la fonction sinc se présente comme celui d'une sinusoide amortie bilatéralement en $1/|u|$ à partir de $u = 0$. Il s'agit du spectre de la fonction valant $1/(2\pi)$ sur $[-\pi, \pi]$ et 0 ailleurs (voir la figure 5).

4.5.2. Interpolation par des polynômes trigonométriques, `dft` et `dct` en 1D et 2D. On rappelle (Lemme 2.9) que, N étant un entier supérieur ou égal à 2 fixé, la transformation linéaire $\texttt{dft}\,(N)$ transforme le vecteur colonne ${}^t[s(0), ..., s(N-1)]$ d'entrées réelles ou complexes en le vecteur colonne

$$ {}^t[\widehat{s}(0), ..., \widehat{s}(N-1)]$$

par multiplication par la matrice $[W_N^{kj}]_{0 \leq k,j \leq N-1}$, k indice de ligne, j indice de colonne, où $W_N := \exp(-2i\pi/N)$. Ceci se lit encore

$$\widehat{s}(k) = \sum_{j=0}^{n-1} s(j)\, e^{-2i\pi jk/N}, \quad k = 0, ..., N-1. \tag{4.39}$$

Le jeu de formules inverses est (voir toujours le Lemme 2.9) :

$$s(j) = \frac{1}{N} \sum_{k=0}^{N-1} \widehat{s}(k)\, e^{2i\pi jk/N}, \quad j = 0, ..., N-1. \tag{4.40}$$

Les formules (4.40) s'interprètent aussi en disant que le polynôme trigonométrique

$$t \in \mathbb{R} \longmapsto \sum_{k=0}^{N-1} \widehat{s(k)}\, e^{\frac{2\pi k}{N}\, i\, t},$$

de fréquences les nombres positifs

$$\omega_{N,0} = 0,\ \omega_{N,1} = 2\pi/N, ..., \omega_{N,N-1} = 2\pi(N-1)/N$$

(uniformément répartis sur $[0, 2\pi[$), est celui qui, bâti à partir de ces fréquences, interpole exactement les valeurs $s(0), ..., s(N-1)$ aux points du maillage entier

$$0 < 1 < ... < N-1$$

(de pas constant égal à 1). Les fréquences de ce polynôme trigonométrique restent dans $[0, 2\pi[$. De fait, comme l'intervalle $[0, 2\pi[$, une fois recentré, devient l'intervalle fréquentiel $[-\pi, \pi[$ et que $\pi/\pi = 1$ est alors le seuil de Nyquist correspondant (prendre $\Omega = \pi$ dans le théorème 4.16 de Shannon-Nyquist), cela n'aurait pas de sens de chercher à interpoler par un polynôme trigonométrique de fréquences hors de $[-\pi, \pi[$ (correspondant à $[0, 2\pi[$ correctement recentré).

Calculer les nombres $\widehat{s}(0), ..., \widehat{s}(N-1)$ suivant les formules (4.40) revient donc à calculer les coefficients de ce polynôme trigonométrique. Plus N est grand, *i.e.* plus l'échantillon est riche, meilleure sera la résolution fréquentielle $2\pi/N$ (le pas du maillage fréquentiel) correspondant à ce polynôme trigonométrique interpolant aux entiers $0, ..., N-1$ les valeurs $s(k)$. On peut forcer cette résolution fréquentielle à être petite en complétant la suite des entrées $[s(0), ..., s(N-1)]$ de manière bilatérale par autant de zéros que l'on veut (`zeropadding`). On a d'ailleurs intérêt à se ramener à travailler avec $N = 2^p$ ($p >> 1$) pour disposer de l'efficacité algorithmique (au niveau du nombre d'opérations, donc du gain en temps de calcul) de l'outil `fft` (voir la Section 2.5).

On peut préférer à la transformation $\texttt{dft}\,(N)$ (qui a le défaut de faire agir une matrice à entrées complexes dès que $N > 2$) la transformation en cosinus $\texttt{cos}_N$ qui transforme le vecteur colonne ${}^t[s(0), ..., s(N-1)]$ de longueur N en le vecteur colonne

$$ {}^t[\texttt{cos}_N\, s\,(0), ..., \texttt{cos}_N\, s\,(N-1)], $$

où

$$ \texttt{cos}_{\rm N}\, s(k) = \alpha_k \sum_{j=0}^{N-1} \cos \frac{\pi k(2j+1)}{2N}\, s(j),\ k = 0, ..., N-1, \tag{4.41} $$

avec $\alpha_0 = \sqrt{1/N}$ et $\alpha_j = \sqrt{2/N}$ si $j \neq 0$. Cette routine est appelée ainsi sous `MATLAB` :

```
cos_N s = dct(s,N);
```

(« *Discrete Cosine Transform* ») et admet une version bi-dimensionnelle (s'appliquant aux tableaux discrets de taille (N_1, N_2) et non plus aux vecteurs colonne de longueur N) transformant un tableau $[I(j_1, j_2)]$, $0 \leq j_1 \leq N_1 - 1$, $0 \leq j_2 \leq N_2 - 1$, en le tableau dont les entrées sont :

$$ \begin{aligned} &\texttt{cos}_{\rm N_1,N_2}\, I\,(k_1, k_2) = \\ &= \alpha_{k_1}\alpha_{k_2} \sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} \cos\Big(\frac{\pi k_1(2j_1+1)}{2N_1}\Big) \cos\Big(\frac{\pi k_2(2j_2+1)}{2N_2}\Big)\, I(j_1, j_2) \end{aligned} \tag{4.42} $$

pour $k_1 = 0, ..., N_1-1$ et $k_2 = 0, ..., N_2-1$. Cette transformation 2D est implémentée sous `MATLAB` en

```
cos_(N_1,N_2)  I = dct2 (I,N_1,N_2);
```

La transformation $\texttt{cos}_N$ s'inverse en

$$ {\texttt{cos}_{\rm N}}^{-1}[ss](j) = \sum_{k=0}^{N-1} \alpha_k \cos\Big(\frac{\pi k(2j+1)}{2N}\Big)\, ss(k),\ j = 0, ..., N, $$

tandis que la transformation $\texttt{cos}_{N_1,N_2}$ s'inverse en

$$\texttt{cos}_{\mathrm{N_1,N_2}}{}^{-1} II(j_1,j_2) = \\ = \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} \alpha_{k_1}\alpha_{k_2} \cos\Big(\frac{\pi k_1(2j_1+1)}{2N_1}\Big) \cos\Big(\frac{\pi k_2(2j_2+1)}{2N_2}\Big) II(k_1,k_2)$$

pour $j_1 = 0, ..., N_1-1$, $j_2 = 0, ..., N_2-1$. (les routines correspondantes à ces routines inverses sous `MATLAB` sont respectivement `idct` et `idct2`.

La transformation `dc2` est très importante en traitement d'image. Les composantes basses-fréquence de l'image I contribuent à la zone située dans l'angle supérieur gauche de $\texttt{cos}\, I$, tandis que les composantes haute-fréquence de I contribuent à la zone située dans l'angle inférieur droit de $\texttt{cos}\, I$ (qui lui est diagonalement opposé), voir la figure 6. Un travail de compression peut être opéré intelligemment sur l'image du tableau I, une fois cette image découpée en blocs de 8×8 pixels : le traitement se fait bloc par bloc ; on met à zéro dans chaque bloc certains coefficients correspondant à des fréquences « moyennes », c'est-à-dire ne contribuant ni au rendu des grosses structures (basses fréquences), ni au rendu des détails (hautes fréquences) de l'image, puis, une fois ce travail fait sur chaque bloc élémentaire 8×8, on revient par $\texttt{dct2}^{-1}$ à une version $\widetilde{I}$ de l'image originelle, mais cette fois compressée : c'est le principe de l'algorithme `JPEG`. D'autres idées (conduisant à des techniques de *tatouage* ou de *stéganographie*) sont basées sur le même principe. Un projet sera justement proposé dans cette direction.

Les transformations `dft` et `dct` sont capitales en traitement des signaux 1D car c'est bien souvent au niveau du spectre d'un signal que sont opérées les transformations les plus utiles (compression, filtrage, tatouage, séparation de sources, coupure des hautes fréquences, *etc.*).

En conclusion, on peut dire que la réalisation de l'interpolation d'une suite de données discrètes par un modèle continu oscillant (se présentant comme un empilement d'ondes) s'avère d'un intérêt capital à la croisée de l'algorithmique numérique et de l'informatique.

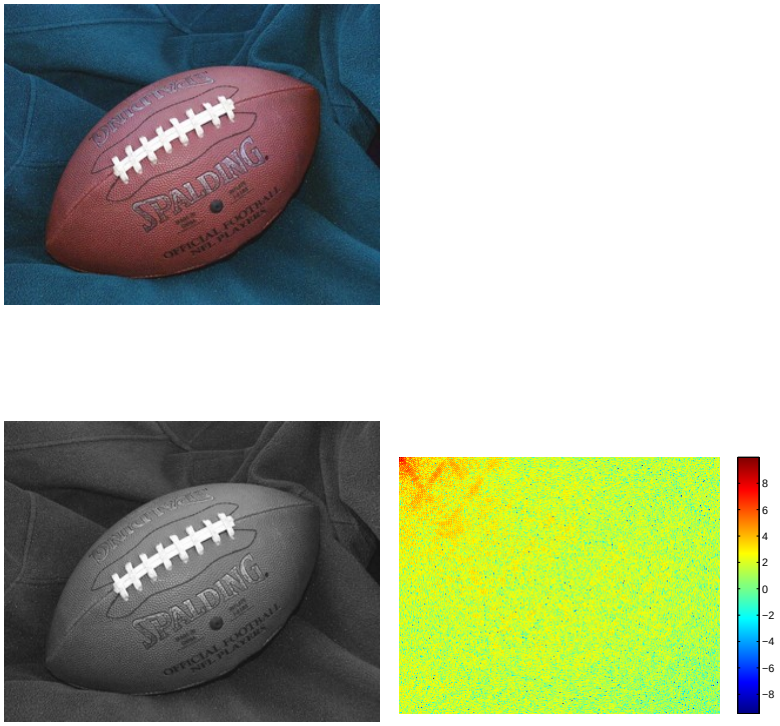

FIGURE 6. Une image et sa transformée en cosinus

-

CHAPITRE 5

Algorithmique numérique et intégration

5.1. Intégration dans $\mathbb{R}^n$, quelques bases pratiques

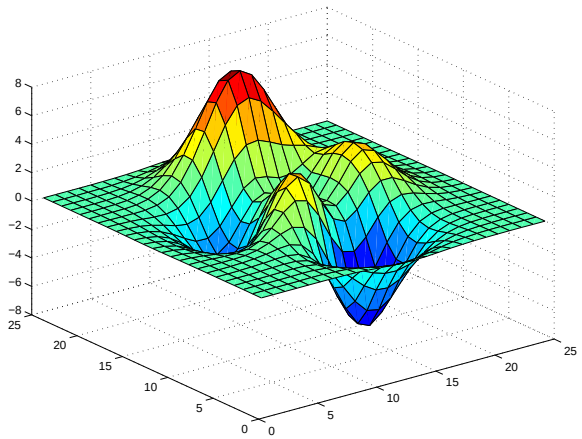

FIGURE 1. Une surface « à coins » et son maillage « remonté sur la surface »

5.1.1. Intégration sur des volumes et des surfaces « à coins ». Dans cette section, on ne considèrera que les sous-ensembles fermés bornés $A \subset \mathbb{R}^n$ se présentant comme l'adhérence (dans $\mathbb{R}^n$) d'un ouvert borné U dont la frontière est un sous-ensemble de $\mathbb{R}^n$ union d'un nombre fini de surfaces $\Sigma_1, ..., \Sigma_N$, chacune de ces surfaces Σ_j pouvant être au voisinage de chacun de ses points x définie localement (c'est-à-dire dans un voisinage U_x de x dans l'espace $\mathbb{R}^n$ ambiant) comme

$$\Sigma_j = \{y \in U_x \, ; \, \sigma_{j,x}(y) = 0\}$$

où $\sigma_{j,x}$ est une fonction de classe C^1 au voisinage de x (dans $\mathbb{R}^n$) telle que $d\sigma_{j,x} \neq 0$ dans ce voisinage.

En chaque point de Σ_j, le vecteur gradient $\vec{\nabla}\sigma_{j,x}$ définit alors un vecteur non nul normal à Σ_j au point x. Ce vecteur induit deux directions possibles : l'une pointant vers l'extérieur de A, dirigée par le vecteur

$$\vec{n}_{\Sigma_j,\text{ext}}(x) := \pm \frac{\vec{\nabla}\sigma_j(x)}{\|\vec{\nabla}\sigma_j(x)\|},$$

l'autre pointant vers l'intérieur de A.

On supposera que les points du bord de A où deux nappes Σ_j et Σ_k (avec $j \neq k$) se coupent forment un sous-ensemble d'intérieur vide dans le bord ∂A de A. Ainsi les

points du bord de A où plusieurs normales extérieures sont envisageables (plusieurs surfaces Σ_j se rencontrant en ces points) constituent un sous-ensemble du bord que l'on pourra considérer comme « négligeable » du point de vue de l'intégration des fonctions sur A. Un tel sous-ensemble A de $\mathbb{R}^n$ sera appelé *domaine de* $\mathbb{R}^n$ *à frontière* C^1 *par morceaux* ou encore plus « prosaiquement » *domaine à coins*. On ne s'interessera dans ce cours qu'à ce type d'ensemble et toutes les fonctions de A dans $\mathbb{R}$ ou $\mathbb{C}$ que nous étudierons seront continues sur l'intérieur de A (on tolère cependant que le module de ces fonctions puisse tendre vers $+\infty$ lorsque l'on se rapproche d'un point du bord de A). Il est évidemment possible de subdiviser un tel ensemble A et d'envisager ainsi des fonctions continues « par morceaux ». On s'intéressera également à des sous-ensembles de la frontière de tels ensembles (que l'on appellera « sufaces à coins » ou « nappes à coins » dans $\mathbb{R}^n$).

L'ensemble A de référence sera pour nous le simplexe de $\mathbb{R}^n$ défini comme

$$\Delta_{n,0} := \Big\{ x \in \mathbb{R}^n \,;\, x_j \geq 0,\ j = 1, ..., n \,;\, \sum_{j=1}^{n} x_j \leq 1 \Big\}.$$

C'est le polyèdre dont les $n+1$ sommets sont l'origine et les extrémités des vecteurs de base. Un « maillage » n-dimensionnel sera pour nous une union finie (dans $\mathbb{R}^n$) de polyèdres fermés δ_j (dits « mailles »), chaque δ_j étant l'image par une application affine inversible de $\mathbb{R}^n$ dans lui-même du simplexe fermé $\Delta_{n,0}$. On exige également les deux contraintes suivantes :

- les intérieurs des mailles δ_j sont disjoints ;
- l'intersection de deux mailles distinctes, si elle est non vide, est une face commune de chacune de ces deux mailles.

Les sommets des polyèdres δ_j sont appelés *nœuds* du maillage.

On se réfèrera aux diverses routines, `surf`, `griddata`, `mesh` de l'environnement `MATLAB` (`3D-Visualization`) pour la visualization de tels domaines à coins *via* le paramétrage (précisément à partir des nœuds d'un maillage) de leur frontière. Le tracé des *surfaces de Bézier* à partir d'une matrice $A = [A_{j_1,j_2}]_{j_1,j_2}$, $0 \leq j_1 \leq M_1$, $0 \leq j_2 \leq M_2$ de points de l'espace *via* le paramétrage

$$\overrightarrow{OM}\,(t,u) = \sum_{j_1=0}^{M_1} \sum_{j_2=0}^{M_2} \binom{M_1}{j_1} \binom{M_2}{j_2} t^{j_1} (1-t)^{M_1-j_1} u^{j_2} (1-u)^{M_2-j_2}\ \overrightarrow{OA_{j_1,j_2}}$$

en fournit une bonne illustration. On pourra réaliser ainsi des surfaces fermées enserrant des domaines « à coins ». On retrouve ces constructions évidemment en Conception Assistée par Ordinateur (CAO) et *Computer Aided Geometric Design*(CAGD).

Exemple 5.1. Dans le cas $n = 1$, on retrouve la notion classique de maillage : étant donné un segment $[a, b]$, une partition de $[a, b]$ en segments fermés induite par une subdivision

$$a = x_0 < x_1 < \cdots < x_N = b.$$

On admettra que si A est un domaine à coins borné, il existe toujours un maillage de $\mathbb{R}^n$ (de mailles $\delta_1, ..., \delta_N$) tel que, pour chaque $j = 1, ..., N$, il existe une fonction φ_j de classe C^1 de δ_j dans $\mathbb{R}^n$ dont le jacobien $\mathrm{jac}(\varphi_j)$ reste strictement positif à

l'intérieur de δ_j, et que l'on ait

$$A = \bigcup_{j=1}^{M} \varphi_j(\delta_j). \tag{5.1}$$

Le polytope de $\mathbb{R}^n$ formé par l'union des δ_j est aussi appelé « *patron* » du domaine à coins A.

Si $\delta_j = L_j(\Delta_{n,0})$ pour $j = 1, ..., M$ et si f est une fonction continue positive sur l'intérieur de A, on définit l'*intégrale* de f sur A comme

$$\begin{aligned} &\int_A f(x_1, ..., x_n)\, dx_1 \cdots dx_n := \\ &= \sum_{j=1}^{M} |\det L_j| \int_{\Delta_{n,0}} \Big(f \circ \varphi_j \circ L_j(\xi)\Big) \Big(\mathrm{jac}[\varphi_j](L_j(\xi))\Big)\, d\xi_1 \cdots d\xi_n \in [0, \infty] \end{aligned} \tag{5.2}$$

en convenant que l'intégrale sur le simplexe $\Delta_{n,0}$ d'une fonction continue positive Φ sur l'intérieur du simplexe $\Delta_{n,0}$ est définie par récurrence sur n par

$$\begin{aligned} &\int_{\Delta_{n,0}} \Phi(\xi_1, ..., \xi_n)\, d\xi_1 \dots d\xi_n = \\ &= \int_{\Delta_{n-1,0}} \Big(\int_0^{1-\sum_{j=1}^{n-1} \xi_j} \Phi(\xi_1, ..., \xi_{n-1}, t)\, dt \Big)\, d\xi_1 \dots d\xi_{n-1} \in [0, \infty]. \end{aligned} \tag{5.3}$$

Une fonction continue sur l'intérieur de A, à valeurs réelles ou complexes, est dite intégrable sur A si

$$\int_A |f(x_1, ..., x_n)|\, dx_1 \dots dx_n < +\infty$$

et l'intégrale de f sur A est alors définie par la formule (5.2). On obtient alors un nombre réel ou complexe, mais en tout cas fini. La définition de l'intégrale ne dépend pas du maillage utilisé pour décrire A.

Si A se présente à partir d'un maillage sous la forme (5.1), la frontière de A se présente comme union finie d'un certain nombre d'ensembles de la forme $\varphi_j(L_j(\gamma_k))$, où $j \in \{1, ..., M\}$ et γ_k est une des $n+1$ faces de dimension $n-1$ du simplexe $\Delta_{n,0}$. Les $n+1$ faces du simplexe $\Delta_{n,0}$ sont paramétrées par les points du simplexe $\Delta_{n-1,0}$:

- la face $\xi_j = 0$, $j = 1, ..., n$ de $\Delta_{n,0}$ est paramétrée par

$$\pi_{\xi_j=0} : (\eta_1, ..., \eta_{n-1}) \in \Delta_{n-1,0} \longmapsto (\eta_1, ..., \eta_{j-1}, 0, \eta_{j+1}, ..., \eta_n) ;$$

- la face $\xi_1 + \cdots + \xi_n = 1$ de $\Delta_{n,0}$ est paramétrée par

$$\pi_{x_1+...+x_n=1} : (\eta_1, ..., \eta_{n-1}) \in \Delta_{n-1,0} \longmapsto (\eta_1, ..., \eta_{n-1}, 1 - \eta_1 - \cdots - \eta_{n-1}).$$

Chaque portion $\sigma = \varphi_j(L_j(\gamma_k))$ du bord de A est ainsi paramétrée par une application γ_σ :

$$\gamma_\sigma : \Delta_{n-1,0} \to \mathbb{R}^n$$

(si $\sigma = \varphi_j \circ L_j(\gamma)$, où γ est une face de $\Delta_{n-1,0}$, on prend $\gamma_\sigma = \varphi_j \circ L_j \circ \pi_\gamma$). Ce paramétrage du bord de A (induit par le maillage) nous permet de définir (et de

calculer) l'intégrale d'une fonction continue h sur ce bord. Si l'on suppose dans un premier temps que la fonction h est continue positive sur ce bord et si

$$\partial A = \bigcup_{\mu=1}^{m} \sigma_\mu,$$

on pose

(5.4)

$$\int_{\partial A} h\, d\sigma_{\partial A} :=$$
$$= \sum_{\mu=1}^{m} \int_{\Delta_{n-1,0}} h(\gamma_{\sigma_\mu}(\eta_1, ..., \eta_{n-1}))\, \sqrt{G_{\gamma_{\sigma_\mu}}(\eta_1, ..., \eta_{n-1})}\, d\eta_1 \dots d\eta_{n-1} \in [0, \infty]$$

où $G_{\gamma_{\sigma_\mu}}$ est le déterminant (toujours positif ou nul) de la matrice $(n-1, n-1)$ des produits scalaires

$$\left\langle \frac{\partial \gamma_{\sigma_l}}{\partial \eta_j}\,,\, \frac{\partial \gamma_{\sigma_l}}{\partial \eta_k} \right\rangle,\ 1 \leq j, k \leq n-1$$

(dite *matrice de Gram* des vecteurs $\partial\gamma_{\sigma_l}/\partial\eta_j$, $j = 1, ...n-1$). Si $h\ : \partial A \to \mathbb{R}$ ou $\mathbb{C}$ est une fonction continue sur δA, on dit que h est intégrable si $|h|$ l'est et on définit l'intégrale de h sur le bord de A toujours par la formule (5.4) (on obtient cette fois un nombre réel ou complexe, mais en tout cas fini). La définition de cette intégrale au bord ne dépend pas du maillage utilisé pour décrire A.

Remarque 5.2. Une partie Σ du bord d'un tel domaine à coins A s'exprimant comme une union finie (mais pas nécessairement tous) d'ensembles du type $\varphi_j(L_j(\gamma_k))$ impliqués dans le « patron » du bord de A est dite *surface à coins* ou *nappe à coins*. Sur une telle nappe, on peut intégrer les fonctions continues suivant la définition (5.4), mais en ne retenant dans la somme que les ensembles $\varphi_j(L_j(\gamma_k))$ impliqués dans la décomposition cellullaire de Σ.

Remarque 5.3 (bord d'une nappe dans $\mathbb{R}^3$ et règle du « bonhomme d'Ampère). Dans le cas $n = 3$, le bord d'une nappe à coins constitue une courbe fermée de $\mathbb{R}^3$ (un « lacet à coins »). Si cette courbe est parcourue dans un sens donné (il y a deux sens possibles pour chaque composante connexe de la courbe), une orientation de la normale extérieure en un point de la portion de nappe ensérrée par une composante connexe de cette courbe est induite par la règle dite *du bonhomme d'Ampère* ou des *trois doigts* : le repère formé par la tangente orientée à la courbe (suivant le mouvement), la normale à la courbe pointant vers la portion de nappe enserrée et la normale extérieure à cette portion de nappe doit être un repère direct (*i.e* de déterminant strictement positif).

5.1.2. La formule de Stokes et ses variantes. La *formule de Stokes* relie, dans le cas où A est domaine à coins, intégration dans A et intégration sur le bord de A. Cette formule d'obédience géométrique est très importante car elle constitue la version multi-dimensionnelle du théorème fondamental de l'analyse : si f est une fonction de classe C^1 au voisinage d'un segment $[a, b)$ de $\mathbb{R}$,

$$\int_{[a,b]} f(x)\, dx = f(b) - f(a). \tag{5.5}$$

On note que le membre de droite de (5.5) peut être interprété comme une intégrale sur le bord de $[a, b]$, $[a, b]$ étant considéré comme un *dipôle* (b chargé positivement,

a chargé négativement). Ceci est cohérent avec le fait que la normale extérieure à $[a,b]$ en b soit dirigée comme la demi-droite $[b,\infty[$ tandis que la normale extérieure à $[a,b]$ en a est dirigée suivant $[a,-\infty[$. Ceci sera précisément en phase avec la formule (5.6) du Théorème de Stokes [1] ci-dessous.

THEORÈME 5.4 (formule de Stokes – ou encore de la divergence –, de Green et de Green-Ostrogradski [2]). *Soit A un domaine à coins borné de $\mathbb{R}^n$ et*

$$x \mapsto \vec{F}(x) = (F_1(x), ..., F_n(x))$$

un champ de vecteurs réel de classe C^1 au voisinage de A. On a alors la formule

$$\int_{\partial A} \langle \vec{F}(x)\,,\, \vec{n}_{\rm ext}(x)\rangle \, d\sigma_{\partial A}(x) = \int_A \mathrm{div}\,(\vec{F})\,(x)\, dx_1 \dots dx_n, \tag{5.6}$$

où la divergence du champ de vecteurs $\vec{F}$ est définie par

$$\mathrm{div}\,(\vec{F})\,(x) := \sum_{j=1}^{n} \frac{\partial F_j}{\partial x_j}(x)$$

En particulier, si $\vec{F} = \vec{\nabla} V$, où $V \ : A \to \mathbb{C}$ désigne un potentiel réel de classe C^2 au voisinage de A, on a la formule de Green :

$$\int_{\partial A} \Big\langle \vec{\nabla} V(x)\,,\, \vec{n}_{\rm ext}(x)\Big\rangle \, d\sigma_{\partial A}(x) = \int_A \Delta\, V\,(x)\, dx_1 \dots dx_n, \tag{5.7}$$

où Δ désigne l'opérateur de Laplace (ou laplacien)

$$\Delta = \sum_{j=1}^{n} \frac{\partial^2}{\partial x_j^2}.$$

Si V_1 et V_2 sont deux potentiels réels de classe C^2 au voisinage de A, cette formule de Green implique aussi la formule de Green-Ostrogradski :

$$\begin{aligned} &\int_{\partial A} \Big\langle V_1(x)\, \vec{\nabla} V_2(x) - V_2(x)\, \vec{\nabla} V_1(x)\,,\, \vec{n}_{\rm ext}(x)\Big\rangle \, d\sigma_{\partial A}(x) = \\ &= \int_A \Big(V_1(x)\, \Delta V_2(x) - V_2(x) \Delta V_1(x)\Big)\, dx_1 \dots dx_n. \end{aligned} \tag{5.8}$$

DÉMONSTRATION. Les formules de Green (5.7) et (5.8) découlent toutes les deux de la formule de la divergence (5.6). En effet $\mathrm{div}\,(\vec{\nabla} V)(x) = \Delta(x)$ pour un potentiel scalaire V de classe C^2. Si V_1 et V_2 sont deux tels potentiels, on vérifie que

$$\mathrm{div}\,\big(V_1(x)\vec{\nabla} V_2(x)\big) = \big\langle \vec{\nabla} V_1(x)\,,\, \vec{\nabla} V_2(x)\big\rangle + V_1(x)\, \Delta V_2(x).$$

Par symétrie, on a aussi

$$\mathrm{div}\,\big(V_2(x)\vec{\nabla} V_1(x)\big) = \big\langle \vec{\nabla} V_2(x)\,,\, \vec{\nabla} V_1(x)\big\rangle + V_2(x)\, \Delta V_1(x).$$

En utilisant la formule de Green deux fois et en faisant la différence, on trouve bien la formule (5.8). Pour prouver la formule de la divergence, on montre que l'on peut se ramener par changement de variables au cas où $A = \Delta_{n,0}$. □

1. George Gabriel Stokes (1819-1903) est un mathématicien et physicien britannique.

2. Au nom de Stokes est ajouté ici celui du mathématicien et physicien russe Mikhail Ostrogradski (1801-1862).

REMARQUE 5.5. Les formules de Green (5.7) et (5.8) jouent un rôle important au niveau de l'informatique (graphisme 3D ou 2D, traitement d'images, Conception Assistée par Ordinateur, Computer Aided Geometric Design). Ce sont elles qui justifient qu'en calculant par exemple le Laplacien d'une image discrète, on aide à en faire surgir les lignes de contraste (correspondant aux contours des objets ou des scènes impliquées dans l'image).

En dimension 3, le bord d'une nappe à coins Σ est un lacet à coins. Un sens de parcours sur le lacet induit une direction de normale extérieure en tout point des diverses portions de nappe enserrées par les composantes connexes du lacet (voir Remarque 5.3). Si l'on dispose d'un champ de vecteurs $\vec{F} := (P, Q, R)$ de classe C^1 au voisinage de Σ, on peut définir la *circulation*[3] du champ sur le lacet $\partial\Sigma$ orienté comme l'*intégrale curviligne*[4]

$$\int_{\partial\Sigma} Pdx + Qdy + Rdz$$

(on exprime x, y, z en termes du paramétrage du lacet orienté). Une conséquence de la formule de Stokes est que l'on a aussi la formule

$$\int_{\partial\Sigma} Pdx + Qdy + Rdz = \iint_{\Sigma} \left\langle \overrightarrow{\text{rot}}\,(\vec{F}(x,y,z)), \vec{n}_{\text{ext}}(x,y,z) \right\rangle d\sigma_{\Sigma}, \tag{5.9}$$

dite *formule de Green-Riemann*. Le *rotationnel* du champ $P\vec{i}+Q\vec{j}+R\vec{k}$ est le champ de vecteurs

$$\overrightarrow{\text{rot}}\ (P\vec{i} + Q\vec{j} + R\vec{k}) := \begin{vmatrix} \frac{\partial}{\partial x} & P & \vec{i} \\ \frac{\partial}{\partial y} & Q & \vec{j} \\ \frac{\partial}{\partial z} & R & \vec{k} \end{vmatrix}$$

Le membre de droite de la formule de Green-Riemann (5.9) s'interprète en termes de physique comme le *flux* du rotationnel du champ $\vec{F}$ au travers de la nappe Σ (orientée en conformité avec le sens de parcours de son bord $\partial\Sigma$ suivant la règle du bonhomme d'Ampère, voir la Remarque 5.3).

5.1.3. La formule de changement de variables ; exemples. La définition de l'intégrale (5.2) (et le fait que cette définition soit indépendante du maillage) repose sur une propriété opérationnelle majeure, la *formule de changement de variables*.

THEORÈME 5.6 (formule de changement de variable). *Soient A et B deux domaines à coins de $\mathbb{R}^n$ tels qu'il existe une application inversible Φ de classe C^1 entre l'intérieur de A et l'intérieur de B. Dire qu'une fonction f continue sur l'intérieur de B est intégrable sur B équivaut à dire que la foncion $(f \circ \Phi) \times |\text{jac}\ (\Phi)|$ est intégrable sur A. De plus, on a la formule*

$$\int_B f(y_1, ..., y_n)\, dy_1 \dots dy_n = \int_A f(\Phi(x_1, ..., x_n))\, |\text{jac}\,(\Phi(x_1, ..., x_n)|\, dx_1 \dots dx_n. \tag{5.10}$$

3. Ceci correspond, en termes de physique, au *travail* du champ de forces lors du parcours du lacet orienté.

4. Voir le cours de MHT401 pour cette notion d'intégrale curviligne sur un chemin paramétré.

Dans le cas $n = 2$, l'un des plus importants changements de variables est celui que traduit le passage du *repérage cartésien* (x, y) au *repérage polaire* (r, θ). Ici $r = \sqrt{x^2 + y^2} \geq 0$ désigne la distance du point (x, y) à l'origine, $\theta \in [0, 2\pi[$ l'angle de vecteurs entre $(1, 0)$ et (x, y). On a les relations

$$x = r\cos\theta, \quad y = r\sin\theta$$

et la formule de changement de variables (5.10) se lit dans ce cadre

$$\iint_{B_{x,y}} f(x, y)\, dx\, dy = \iint_{A_{r,\theta}} f(r\cos\theta, r\sin\theta)\, r\, dr\, d\theta. \tag{5.11}$$

Si elle s'avère une transformation simple du point de vue mathématique, le passage du repérage cartésien au repérage polaire pose de gros problèmes au niveau de l'informatique graphique. En effet, passer d'un maillage cartésien au maillage polaire soulève des questions d'interpolation : si l'on fait pivoter un pixel (x, y) de ce maillage autour de l'origine d'un angle donné θ, le nouveau point obtenu n'est plus évidemment un pixel du maillage cartésien initial et se doit donc d'être interpolé : on peut choisir le pixel le plus proche, interpoler entre les quatre pixels sommets du carré dans lequel on tombe, *etc.* Sous `MATLAB` par exemple, la commande (fort utile) `imrotate`, avec ses diverses options (`'nearest'`, `'bilinear'`, `'bicubic'`), traduit bien ces difficultés.

Dans l'espace $\mathbb{R}^3$, le *repérage sphérique* des points (x, y, z) (en coordonnées cartésiennes) s'effectue à partir de la distance à l'origine

$$r = \sqrt{x^2 + y^2 + z^2},$$

de la *longitude* $\theta \in [0, 2\pi[$ (le plan $x0z$ étant considéré comme le plan méridien de référence, tel le plan méridien de Greenwich), enfin de la *colatitude* $\varphi \in [0, \pi/2]$, angle des vecteurs $(0, 0, 1)$ et (x, y, z) (c'est la latitude, mais repérée non plus à partir de l'équateur, mais cette fois du pôle Nord). Les angles θ et φ sont appelés *angles d'Euler*. On a les relations

$$x = r\sin\varphi\cos\theta\,, \quad y = r\sin\varphi\sin\theta\,, \quad z = r\cos\varphi$$

et la formule de changement de variables (5.10) se lit dans ce cadre

$$\begin{aligned} &\iiint_{B_{x,y,z}} f(x, y, z)\, dx\, dy\, dz = \\ &= \iiint_{A_{r,\theta,\varphi}} f(r\sin\varphi\cos\theta, r\sin\varphi\sin\theta, r\cos\varphi)\, r\, \sin\varphi\, dr\, d\theta\, d\varphi. \end{aligned} \tag{5.12}$$

Le passage d'un maillage cartésien à un maillage sphérique pose du point de vue informatique (en pire) les mêmes problèmes que ceux que pose en dimension 2 le passage du maillage cartésien au maillage polaire.

Toujours dans $\mathbb{R}^3$, on peut utiliser aussi le *repérage cylindrique*. Le point (x, y, z) est repéré par sa distance $r = \sqrt{x^2 + y^2} \geq 0$ à l'axe $z'Oz$, sa longitude θ et son *altitude* (ou sa « cote ») z. Les formules sont

$$x = r\cos\theta\,, \quad y = r\sin\theta\,, \quad z = z$$

et la formule de changement de variables (5.10) se lit dans ce cadre

$$\iiint_{B_{x,y,z}} f(x, y, z)\, dx\, dy\, dz = \iiint_{A_{r,\theta,z}} f(r\cos\theta, r\sin\theta, z)\, r\, dr\, d\theta\, dz. \tag{5.13}$$

5.1.4. Intégration « par tranches » ; le théorème de Fubini. La méthode inductive utilisée pour intégrer les fonctions continues sur le simplexe (voir (5.3)) induit (par changement de variables et passage *via* un maillage) une extension dans un cadre général.

Supposons que les n coordonnées $(x_1, ..., x_n)$ de $\mathbb{R}^n$ soient organisées en deux blocs $x' = (x_1, ..., x_k)$ et $x'' = (x_{k+1}, ..., x_n)$, $1 \leq k \leq n-1$. Si A est un domaine à coins dans $\mathbb{R}^n$, alors la projection $\mathrm{proj}_{x'}(A)$ de A sur l'espace $\mathbb{R}^k_{x_1,...,x_k}$ est un domaine à coins de $\mathbb{R}^k$. Pour chaque point x' dans $\mathrm{proj}_{x'}(A)$, la « tranche » de A au dessus de x',

$$A^{x'} := \{y \in \mathbb{R}^{n-k}, \, ; \, (x', y_1, ..., y_{n-k}) \in A\}$$

est un domaine à coins de $\mathbb{R}^{n-k}$. On peut alors énoncer la formule permettant de toujours ramener le calcul numérique des intégrales sur des volumes ou des surfaces à des calculs d'intégrales de fonctions d'une variable.

THEORÈME 5.7 (théorème de Fubini, version opérationnelle). *Si f est une fonction intégrable sur un domaine à coins de $\mathbb{R}^n$, on a*

$$\begin{aligned} &\int_A f(x_1, ..., x_n)\, dx_1 \dots dx_n = \\ &\int_{\mathrm{proj}_{x'}(A)} \Big(\int_{A^{x'}} f(x', y_1, .., y_{n-k})\, dy_1 \dots dy_{n-k} \Big)\, dx_1 \dots dx_k. \end{aligned} \tag{5.14}$$

5.2. Les méthodes de Newton-Cotes, de quadrature et composites

Dans cette section, nous allons détailler (dans le cas de la dimension 1, cas auquel, on l'a vu dans la section précédente, les calculs se ramènent toujours d'après le Théorème 5.7) le calcul approché d'intégrales de fonctions continues sur un segment $[a, b]$ (voire un intervalle non borné (a, b) modulo certaines précautions assurant l'intégrabilité des fonctions) sur lequel on dispose d'un maillage

$$a \leq x_0 < x_1 < \cdots < x_N \leq b. \tag{5.15}$$

5.2.1. Présentation des méthodes de Newton-Cotes. L'objectif que nous avons ici est de présenter un calcul approché de l'intégrale

$$I[f; [a, b]] := \int_a^b f(t)\, dt$$

(f étant une fonction continue sur un segment borné $[a, b]$) avec ces deux exigences :

- l'intégrale approchée $I_{\mathrm{app}}[f; [a, b]]$ sur $[a, b]$ dépend de manière linéaire des entrées évaluations de f aux $N + 1$ nœuds du maillage, *i.e.* $f(x_0), ..., f(x_N)$;
- le calcul approché devient exact, c'est-à-dire

  $$I_{\mathrm{app}}[f; [a, b]] = I[f; [a, b]],$$

  lorsque f est une fonction polynomiale de degré inférieur où égal à N.

On souhaite donc déterminer des scalaires $\lambda_0, ..., \lambda_N$ (« universelles », c'est-à-dire ne dépendant que des nœuds $x_0, ..., x_N$ du maillage) tels que

$$I_{\mathrm{app}}[f; [a, b]] = \sum_{j=0}^{N} \lambda_j\, f(x_j) \tag{5.16}$$

et

(5.17)
$$\int_a^b t^k\,dt = \frac{b^{k+1}-a^{k+1}}{k+1} = I[t^k;[a,b]] = I_{\rm app}[t^k;[a,b]] = \sum_{j=0}^{N} \lambda_j\, x_j^k \quad \forall\, k = 0,...,N.$$

Comme les x_j, $j = 0,...,N$, sont $N+1$ points distincts, le système linéaire de $N+1$ équations en les $N+1$ inconnues $\lambda_0,...,\lambda_N$ est de Cramer (le déterminant de ce système est ce que l'on appelle un déterminant de Vandermonde, valant $\prod_{0\le j_1<j_2\le N}(x_{j_2}-x_{j_1})$, donc non nul puisque les x_j sont distincts). Il existe donc un unique vecteur de coefficients $(\lambda_0,...,\lambda_N)$ solution de notre problème. Voici plusieurs cas particuliers importants.

– Le cas où $N = 0$ et où, par souci de compromis, on prend $x_0 = (a+b)/2$. On trouve dans ce cas $\lambda_0 = b-a$ et la formule approchée est dans ce cas

(5.18)
$$I_{\rm app}[f;[a,b]] = (b-a) f\Big(\frac{a+b}{2}\Big).$$

Ce calcul approché est dit *méthode des rectangles.*. Il se trouve que l'on a de la chance ici car la formule

(5.19)
$$\int_a^b f(t)\,dt = \mathrm{I}_{\rm app}[f;[a,b]]$$

se révèle exacte pour les fonctions polynomiales de degré 1 (elle devient fausse par contre pour les fonctions polynomiales de degré 2).

– Le cas où $N = 1$ et où, toujours par souci de compromis, on prend $x_0 = a$ et $x_1 = b$. Dans ce cas, on trouve $\lambda_0 = \lambda_1 = (b-a)/2$ et la formule approchée devient

(5.20)
$$I_{\rm app}[f;[a,b]] = \frac{b-a}{2}\,(f(a)+f(b)).$$

Ce calcul approché correspond à la *méthode des trapèzes*. Le calcul approché cesse aussi d'être exact pour les fonctions polynomiales de degré 2.

– Le cas $N = 2$ où, toujours par souci de compromis, on prend $x_0 = a$, $x_1 = (a+b)/2$ et $x_2 = b$. On trouve dans ce cas

$$\lambda_0 = \lambda_2 = \frac{b-a}{6} \quad \& \quad \lambda_1 = \frac{2(b-a)}{3}$$

(il est facile de résoudre le système de Cramer (5.17) dans ce cas) et la formule approchée devient

(5.21)
$$I_{\rm app}[f;[a,b]] = \frac{b-a}{6}\,\Big(f(a) + 4\,f\Big(\frac{a+b}{2}\Big) + f(b)\Big).$$

Cette méthode est dite *méthode de Simpson*[5]. Un miracle se produit encore ici : la formule (5.19) est encore valide (comme on le vérifie aisément) pour les fonctions polynomiales de degré 3 (elle est fausse par contre pour les fonctions polynomiales de degré 4).

5. Ainsi dénommée en référence au mathématicien et astrologue britannique Thomas Simpson (1710-1761) ; de fait, elle avait été déjà introduite par Johannes Kepler deux siècles auparavant.

– Si $N = 3$, on introduit, toujours par souci de symétrie, les 4 points $x_0 = a$, $x_1 = a + (b-a)/3$, $x_2 = a + 2(b-a)/3$, $x_3 = b$. Le calcul des coefficients (facile à mener en résolvant un système de Cramer à 4 inconnues) conduit à

$$\lambda_0 = \lambda_3 = \frac{b-a}{8} \quad \& \quad \lambda_1 = \lambda_2 = 3\,\frac{b-a}{8}.$$

La formule approchée est donc dans ce cas

$$I_{\rm app}[f;[a,b]] = \frac{b-a}{8}\Big(f(a) + 3\Big(f\Big(\frac{2a+b}{3}\Big) + f\Big(\frac{a+2b}{3}\Big)\Big) + f(b)\Big). \tag{5.22}$$

C'est la *méthode à quatre points*. La formule (5.19) cesse d'être vraie pour les fonctions polynomiales de degré 4 et au delà.

Toutes les méthodes présentées ici, dans lesquelles le maillage est un maillage régulier (ou encore à pas constant) entrent dans la catégorie des méthodes dites de *méthodes de Newton-Cotes*[6].

5.2.2. Méthodes de Gauss ou de quadrature. On peut oublier le souci de symétrie (et de compromis) en envisageant des maillages qui ne soient pas uniformément répartis. Ce que nous avons vu dans la section 4.2.3 à propos du problème de la meilleure approximation uniforme (et du rôle des polynômes de Tchebychev du fait du Théorème 4.9 d'alternance) nous conforte en effet dans cette voie. Supposons que l'intégrale à calculer de manière approchée à partir du maillage, c'est-à dire sous la forme

$$I_{\omega,{\rm app}}[f;[a,b]] = \sum_{j=0}^{N} \lambda_j\, f(x_j), \quad \lambda_0, ..., \lambda_N \in \mathbb{R} \text{ (universelles en fonction des } x_j)$$

soit une intégrale « pondérée »

$$\int_a^b f(t)\,\omega(t)\,dt = I_\omega[f;(a,b)],$$

où $\omega : (a,b) \to [0,\infty]$ soit un poids comme dans la section 4.3.1, avec $-\infty \le a < b \le +\infty$. On considère, associé à ce poids, le système $\{P_k\}_{k=0,1,...}$ de polynômes unitaires (avec $\deg P_k = k$), orthonormé relativement au produit scalaire

$$\langle f, g\rangle := \int_{(a,b)} f(t)\,\overline{g(t)}\,\omega(t)\,dt, \tag{5.23}$$

qui a été construit *via* le procédé de Gram-Schmidt dans la Section 4.3.1. Fixons $N \ge 0$ et prenons pour $x_0, ..., x_N$ les $N+1$ zéros (tous réels) du polynôme P_{N+1} (tous dans (a,b), voir la Section 4.3.1). Il existe des constantes réelles $\lambda_0, ..., \lambda_N$ uniques telles que, pour toute fonction polynomiale de degré inférieur où égal à N, pour toute fonction continue sur (a,b) telle que

$$\int_{(a,b)} |f(t)|^2\,\omega(t)\,dt < +\infty,$$

on ait la formule

$$\int_{(a,b)} f(t)\,\omega(t)\,dt = I_\omega[f;[a,b]] = I_{\omega,{\rm app}}[f;[a,b]] = \sum_{j=0}^{N} \lambda_j\, f(x_j).$$

6. Ces formules sont apparues l'occasion du travail de relecture par le mathématicien anglais Roger Cotes (1682-1716) des *Principia* d'Isaac Newton.

Comme l'espace vectoriel engendré par les restrictions à (a,b) des fonctions $t \mapsto t^k$, $k = 0, ..., N+1$, est le même que celui engendré par les restrictions à (a,b) des fonctions $t \mapsto P_k(t)$, chercher les λ_j revient à résoudre le système (qui est encore de Cramer car les x_j sont distincts et que P_k est de degré exactement k pour tout k)

$$\sum_{j=0}^{N} \lambda_j \, P_k(x_j) = \int_{(a,b)} P_k(t)\, \omega(t)\, dt = \begin{cases} \int_a^b \omega(t)\, dt & \text{si} \quad k = 0 \\ 0 \quad \text{si} \quad k = 1, ..., N+1. \end{cases} \tag{5.24}$$

Une fois ces λ_j choisis, on constate que la formule

$$\int_{(a,b)} f(t)\, \omega(t)\, dt = \sum_{j=0}^{N} \lambda_j f(x_j) \tag{5.25}$$

(conçue pour être exacte pour toutes les fonctions polynomiales f de degré au plus N) l'est en fait pour toutes les fonctions polynomiales de degré au plus $2N+1$, ce qui fournit un gain notoirement appréciable en termes (on le verra) du calcul d'erreur. En effet, si P est un polynôme de degré au plus $2N+1$, on peut utiliser la division euclidienne et écrire

$$P(X) = P_{N+1}(X)Q(X) + R(X) \quad \text{avec} \quad \deg R \leq N. \tag{5.26}$$

En fait, on a aussi $\deg Q \leq N$ car $\deg P \leq 2N+1$ et $\deg P_{N+1} = N+1$. En intégrant (5.26), on trouve

$$\int_{(a,b)} P(t)\, \omega(t)\, dt = \int_{(a,b)} P_{N+1}(t)\, Q(t)\, \omega(t)\, dt + \int_{(a,b)} R(t)\, \omega(t)\, dt.$$

Or

$$\int_{(a,b)} P_{N+1}(t)\, Q(t)\, \omega(t)\, dt = 0$$

puisque P_{N+1} est par construction orthogonal (relativement au produit scalaire (5.23)) à $P_0, ..., P_N$, donc à toutes les fonctions polynomiales de degré au plus N, donc à Q. D'autre part

$$\int_{(a,b)} R(t)\, \omega(t)\, dt = \sum_{j=0}^{N} \lambda_j R(x_j)$$

puisque $\deg R \leq N$ et que la formule (5.25) est présisément faite pour être exacte pour toutes les fonctions polynomiales de degré au plus N. Comme

$$P(x_j) = P_{N+1}(x_j)\, Q(x_j) + R(x_j) = 0 \times Q(x_j) + R(x_j) = R(x_j)$$

pour $j = 0, ..., N$, on a

$$\int_{(a,b)} R(t)\, \omega(t)\, dt = \sum_{j=0}^{N} \lambda_j R(x_j) = \sum_{j=0}^{N} \lambda_j P(x_j),$$

ce qui prouve l'exactitude de la formule (5.25) lorsque $f = P$, P étant une fonction polynomiale de degré $2N+1$. Ce type de méthode d'intégration basée sur l'orthogonalité relativement à un poids permettant sans frais de réaliser une formule approchée devenant exacte pour les fonctions polynomiales de degré jusqu'à $2N+1$ lorsque le maillage est un maillage à seulement $N+1$ nœuds (mais par contre bien

choisis !) est dit méthode de Gauss [7] ou *méthode de quadrature* (en référence à la méthode des « moindres carrés »).

5.2.3. Ordre d'exactitude et contrôle d'erreur.

DÉFINITION 5.8 (exactitude à un ordre précisé). Une formule d'intégration approchée du type

$$I_{\text{app}}[f;[a,b]] \simeq I[f;[a,b]] \quad ou \quad I_{\omega,\text{app}}[f;(a,b)] \simeq I_\omega[f;(a,b)]$$

est dite *exacte à l'ordre p* si elle est exacte pour toute fonction polynomiale de degré au plus p et ne l'est pas pour la fonction $t \mapsto t^{p+1}$.

EXEMPLE 5.9. Les méthodes des trapèzes ou des rectangles sont exactes à l'ordre 1. Celles de Simpson et des trois points sont exactes à l'ordre 3. La méthode de quadrature à $N+1$ points est exacte à l'ordre $2N+1$.

Considérons d'abord les méthodes du type Newton-Cotes. Pour contrôler l'erreur dans une telle méthode, on utilise la formule de Taylor avec reste intégral (en a) (Proposition 3.3) qui assure que, si f est de classe C^∞, on peut écrire

$$f(x) = \text{Taylor}_p[f;a](x) + \frac{1}{p!}\int_a^x (x-t)^p f^{(p+1)}(t)\,dt,$$

où $\text{Taylor}_p[f;a]$ est le polynôme de Taylor de f à l'ordre p en a, soit

$$\text{Taylor}_p[f;a] = \sum_{k=0}^{p} \frac{f^{(k)}(a)}{k!}\,(x-a)^k.$$

Si on note $(x-t)_+ := \sup(x-t,0)$, l'erreur $E[f;[a,b]]$ commise entre $I[f;[a,b]]$ et $I_{\text{app}}[f;[a,b]]$ dans une formule de Newton-Cotes est celle que l'on commet avec la fonction

$$x \in [a,b] \mapsto \frac{1}{p!}\int_a^b (x-t)_+^p f^{(p+1)}(t)\,dt.$$

Cette erreur s'exprime aussi (si l'on utilise le théorème de Fubini) comme

$$E[f;[a,b]] = \frac{1}{p!}\int_a^b f^{(p+1)}(t)\,E\Big[x \mapsto (x-t)_+^p;[a,b]\Big]\,dt$$

En particulier, en utilisant la formule de la moyenne [8], on trouve, si la fonction

$$t \in [a,b] \longmapsto E\Big[x \mapsto (x-t)_+^p\Big]$$

garde un signe constant [9], que

$$E[f;[a,b]] = \frac{f^{(p+1)}(\xi)}{p!}\int_a^b E\Big[x \mapsto (x-t)_+^p;[a,b]\Big]\,dt.$$

---

7. On retrouve ici le mathématicien, astronome et philosophe allemand Carl Friedrich Gauss (1777-1855), certainement l'un de ceux qui ont le plus contribué à guider l'évolution des mathématiques tous domaines confondus (algèbre, analyse, théorie des nombres et géométrie).

8. Qui assure $\int_a^b u(t)\,v(t)\,dt = u(\xi)\int_a^b v(t)\,dt$ pour un certain $\xi \in [a,b]$ si u et v sont continues sur $[a,b]$ et v y garde un signe constant.

9. Ce sera le cas dans nos exemples, on le verra.

Le calcul de $E[f:[a,b]]$ (dans tous les cas de figure pour une méthode de Newton-Cotes exacte à l'ordre p) conduit (pour une fonction C^∞ sur $[a,b]$) à une estimation d'erreur du type

$$|E[f;[a,b]]| = \Big|I[f;[a,b]] - I_{\text{app}}[f;[a,b]]\Big| \leq C[f] \times (b-a)^{p+2}. \tag{5.27}$$

Ceci résulte du double processus d'intégration impliqué dans le calcul de l'expression

$$\int_a^b E\Big[x \mapsto (x-t)_+^p; [a,b]\Big]\, dt$$

dans laquelle figure déjà sous le premier intégrant la fonction $x \mapsto (x-t)_+^p$. Les calculs peuvent être menés explicitement dans les trois premiers exemples mentionnés (rectangles et trapèzes avec $p=1$, Simpson avec $p=3$) :

– pour la méthode des rectangles, on trouve, pour $t \in [a,b]$,

$$E_{\text{rect}}\Big[x \mapsto (x-t)_+^p; [a,b]\Big] = \begin{cases} \frac{(t-a)^2}{2} \text{ si } t \leq \frac{a+b}{2} \\ \frac{(t-b)^2}{2} \text{ si } t \geq \frac{a+b}{2} \end{cases}$$

et, en intégrant sur $[a,b]$, puis en appliquant la formule de la moyenne

$$E_{\text{rect}}[f;[a,b]] = f''(\xi) \times \frac{(b-a)^3}{24} \quad (\xi \in [a,b]). \tag{5.28}$$

– pour la méthode des trapèzes, on trouve, pour $t \in [a,b]$,

$$E_{\text{trap}}\Big[x \mapsto (x-t)_+^p; [a,b]\Big] = \frac{(t-a)(t-b)}{2}$$

et donc, en intégrant sur $[a,b]$, puis en appliquant la formule de la moyenne

$$E_{\text{trap}}[f;[a,b]] = -f''(\xi) \times \frac{(b-a)^3}{12} \quad (\xi \in [a,b]). \tag{5.29}$$

– pour enfin la méthode de Simpson, les calculs sont plus laborieux mais l'on trouve, pour $t \in [a,b]$,

$$E_{\text{Simp}}\Big[x \mapsto (x-t)_+^p; [a,b]\Big] = \begin{cases} \frac{(b-t)^3(2b+a-3t)}{12} \text{ si } t \leq \frac{a+b}{2} \\ \frac{((a-t)^3(b+2a-3t)}{2} \text{ si } t \geq \frac{a+b}{2} \end{cases}$$

et donc, en intégrant sur $[a,b]$, puis en appliquant la formule de la moyenne

$$E_{\text{Simp}}[f;[a,b]] = -f^{(4)}(\xi) \times \frac{(b-a)^5}{2880} \quad (\xi \in [a,b]). \tag{5.30}$$

Pour une méthode à quadrature à $N+1$ nœuds (donc exacte à l'ordre $2N+1$), on raisonne différement pour contrôler l'erreur. On introduit le polynôme H_{N+1} d'interpolation de Hermite (de degré $2(N+1)-1 = 2N+1$) interpolant les valeurs de la fonction f et de ses dérivées aux $N+1$ nœuds $x_0, ..., x_N$ du maillage (les zéros du polynôme unitaire P_{N+1} de la famille de polynômes orthogonaux relativement au poids ω). On utilise le fait (la preuve est analogue à celle de la Proposition 3.4) que

$$f(t) - H_{N+1}(t) = \frac{f^{2(N+1)}(\xi_t)}{(2(N+1)!} \prod_{j=0}^{N} (t-x_j)^2 = \frac{f^{2(N+1)}(\xi_t)}{(2(N+1)!} P_{N+1}^2(t).$$

On en déduit

$$I_\omega[f;(a,b)] - I_{\omega,\mathrm{app}_N}[f;(a,b)] = I_\omega[f;(a,b)] - I_\omega[H_{N+1};(a,b)] = \\ = \frac{f^{(2(N+1))}(\xi)}{(2(N+1))!}\int_{(a,b)} P_{N+1}^2(t)\,\omega(t)\,dt.$$

En en déduit donc que l'erreur dans la méthode de Gauss pour un poids ω (avec $N+1$ nœuds) est donnée par :

(5.31)

$$E_\omega[f;(a,b)] = I_\omega[f;(a,b)] - I_{\omega,\mathrm{app}_N}[f:(a,b)] = \frac{f^{(2(N+1))}(\xi)}{(2(N+1))!}\int_{(a,b)} P_{N+1}^2(t)\,\omega(t)\,dt.$$

Lorsque (a,b) est un segment $[a,b]$, la quantité $b-a$ ne figure pas explicitement, mais elle est implicite derrièe l'intégrale

$$\int_{(a,b)} P_{N+1}^2(t)\,\omega(t)\,dt.$$

On retrouve une estimation en $(b-a)^{2(N+1)+1} = (b-a)^{2N+3} = (b-a)^{p+2}$, ce qui est en accord avec le fait que la méthode est exacte à l'ordre $p = 2N+1$.

5.2.4. Les méthodes « composites » et la méthode de Romberg. Pour profiter des estimations d'erreur en $C[f]\times(b-a)^{p+2}$ (lorsque la méthode est exacte à l'ordre p), il faut évidemment que $b-a << 1$ (sinon le contrôle d'erreur explose). Il convient donc, pour calculer une intégrale

$$\int_a^b f(t)\,dt \quad \text{ou} \quad \int_a^b f(t)\,\omega(t)\,dt$$

lorsque $-\infty < a < b < +\infty$, de découper $[a,b]$ en N sous-intervalles de longueur $h = (b-a)/N$ et d'appliquer à chaque intervalle de la subdivision ainsi obtenue une des méthodes de calcul approché (rectangles, trapèzes, Simpson, Gauss, ...). Ce procédé est dit *méthode composite* ou *méthode hybride*.

Si la méthode $\mathcal{M}$ utilisée pour le calcul approché dans chaque sous-intervalle est exacte à l'ordre p, la majoration d'erreur dans la méthode « composite » qui en résulte (consistant à utiliser $\mathcal{M}$ dans chacun des N intervalles de la subdivision) est contrôlée en module par

$$|E_{\mathrm{comp}}^{[N]}[f;[a,b]]| \le C[f]\times N\times\Big(\frac{b-a}{N}\Big)^{p+2} = O(N^{-p-1}). \tag{5.32}$$

La formule d'Euler-MacLaurin dans sa version continue (Proposition 3.6) se lit aussi (voir (3.19)), $[a,b]$ désignant un segment, f une fonction de classe C^∞ sur $[a,b]$,

$h = (b-a)/N$, m un ordre arbitraire,

(5.33)

$$
\begin{aligned}
&h\Big(\frac{f(a)}{2} + \sum_{k=1}^{N-1} f(a+kh) + \frac{f(b)}{2}\Big) = \\
&= \int_a^b f(t)\,dt + \sum_{l=1}^{m} \frac{b_{2l}}{(2l)!}\left(f^{(2l-1)}(b) - f^{(2l-1)}(a)\right) h^{2l} + \\
&\quad + \frac{h^{2m}}{(2m)!}\int_a^b B_{2m}\Big(\frac{t-a}{h} + 1 - E\Big[\frac{t-a}{h}+1\Big]\Big)\, f^{(2m)}(t)\,dt \\
&= \int_a^b f(t)\,dt - \Big(\frac{f'(a)-f'(b)}{12}\Big) h^2 + \Big(\frac{f^{(3)}(b)-f^{(3)}(a)}{720}\Big) h^4 + \cdots + O(h^{2m}).
\end{aligned}
$$

On remarque ainsi, si $h_N = (b-a)/2^N$, avec $N \in \mathbb{N}^*$, que

$$
h_N\Big(\frac{f(a)}{2} + \sum_{k=1}^{2^N-1} f(a+kh_N) + \frac{f(b)}{2}\Big)
$$

représente la valeur approchée de l'intégrale $I[f;[a,b]]$ calculée par la méthode composite construite à partir de la méthode des trapèzes (avec le pas h_N). Notons cette quantié $I_N^{[1]}$. On remarque que

$$
\begin{aligned}
I_N^{[1]} &= \int_a^b f(t)\,dt - \Big(\frac{f'(a)-f'(b)}{12}\Big) h_N^2 + \Big(\frac{f^{(3)}(b)-f^{(3)}(a)}{720}\Big) h_N^4 + \cdots \\
I_{N+1}^{[1]} &= \int_a^b f(t)\,dt - \Big(\frac{f'(a)-f'(b)}{12}\Big) \frac{h_N^2}{4} + \Big(\frac{f^{(3)}(b)-f^{(3)}(a)}{720}\Big) \frac{h_N^4}{16} + \cdots
\end{aligned}
$$

En combinant ces deux relations, il vient

$$
\frac{4I_{N+1}^{[1]} - I_N^{[1]}}{3} = \int_a^b f(t)\,dt + O(h_N^4). \tag{5.34}
$$

On constate d'ailleurs que

$$
I_N^{[2]} := \frac{4I_{N+1}^{[1]} - I_N^{[1]}}{3}
$$

n'est rien d'autre que le calcul approché de $I[f;[a,b]]$ par la méthode composite batie cette fois sur la méthode de Simpson (avec le pas h_N). L'approximation de l'intégrale est ici en $O(h_N^4)$ au lieu de $O(h_N^2)$. Ce procédé d'accélération de convergence peut être itéré car on peut profiter de toute la force de la formule d'Euler-Maclaurin. On pose

$$
I_N^{[k+1]} := \frac{4I_{N+1}^{[k]} - I_N^{[k]}}{3}, \quad k = 1, 2, .. \tag{5.35}
$$

On a

$$
I_N^{[k]} = I[f;[a,b]] + O(h_N^{2k}).
$$

Cette méthode (inspirée de l'algorithme de Richardson, voir la Section 2.6.2) est dite *méthode de Romberg*[10].

10. Du nom du mathématicien allemand Werner Romberg (1909-2003) qui l'introduisit dans ses travaux en intégration numérique.

CHAPITRE 6

Equations Différentielles Ordinaires (EDO)

6.1. Les bases théoriques : Cauchy-Lipschitz

On se propose de modéliser un phénomène physique $t \longmapsto Y(t)$ (et, si possible, d'en anticiper le passé ou d'en prévoir l'évolution à partir de sa valeur $Y_0 = Y(t_0)$ à l'instant $t = t_0$). Ici $(t, Y(t))$ prend ses valeurs dans un ouvert U de $\mathbb{R}^{n+1}$ dit *espace des phases* ou encore *espace des états* du phénomène. On fait l'hypothèse que ce phénomène est régi (on dit aussi « contraint ») par une équation différentielle

$$Y'(t) = F(t, Y(t))\,, \tag{6.1}$$

F désignant une fonction continue dans l'ouvert U et à valeurs dans $\mathbb{R}^n$. Ce qui signifie que $t \longmapsto Y(t)$ est de classe C^1 sur son intervalle ouvert I de vie (à déterminer) autour de t_0, et se plie sur cet intervalle I à la relation (6.1).

Le théorème majeur que nous admettrons ici (et qui soutend de fait la résolution numérique du problème) est le Théorème de Cauchy-Lipschitz [1] dont voici l'énoncé.

THEORÈME 6.1 (théorème de Cauchy-Lipschitz). *Soit U un ouvert de $\mathbb{R}^{n+1}$ et $F : U \to \mathbb{R}^n$ une fonction continue satisfaisant au voisinage de tout point la condition suivante (dite de Lipschitz) : pour tout $(t_0, Y_0) \in U$, il existe $\epsilon > 0$, $\eta > 0$, $K \geq 0$ (dépendants (t_0, Y_0)) tels que $[t_0 - \epsilon, t_0 + \epsilon] \times \overline{B_{\mathbb{R}^n}(Y_0, \eta)} \subset U$ et*

$$\begin{aligned} &\forall\, t \in [t_0 - \epsilon, t_0 + \epsilon]\,,\ \forall\, Y_1, Y_2 \in \overline{B_{\mathbb{R}^n}(Y_0, \eta)}, \\ &\|F(t, Y_1) - F(t, Y_2)\| \leq K\, \|Y_1 - Y_2\|. \end{aligned} \tag{6.2}$$

Alors, pour tout $(t_0, Y_0) \in U$, il existe un <u>unique</u> couple (I, Y), où I est un intervalle ouvert de $\mathbb{R}$, $Y : I \to \mathbb{R}$ une fonction de classe C^1, tel que :

(1) *pour tout $t \in I$, $(t, Y(t)) \in U$ et*

$$\begin{aligned} &Y(t_0) = Y_0 \ (\textit{condition initiale}) \\ &\forall\, t \in I, \quad (t, Y(t)) \in U \quad \& \quad Y'(t) = F(t, Y(t)) \end{aligned} \tag{6.3}$$

(on dit que (I, Y) est solution du problème de Cauchy (6.3));

(2) *si $(\widetilde{I}, \widetilde{Y})$ est un autre couple solution du même probème de Cauchy (6.3), alors $\widetilde{I} \subset I$ et $\widetilde{Y}$ est la restriction de Y à $\widetilde{I}$ (on dit que (I, Y) est une solution maximale du problème de Cauchy (6.3)).*

Dans le contexte de l'algorithmique numérique, nous retiendrons un résultat plus fort (avec des hypothèses plus contraignantes). Si $U = I_0 \times \mathbb{R}^n$ et $F : I_0 \times \mathbb{R}^n \to \mathbb{R}^n$

1. Au nom du mathématicien français Augustin Cauchy (1789-1857) est ici associé celui de l'analyste allemand Rudolph Lipschitz (1832-1903), à qui l'on doit la mise en évidence de l'importance de la condition (6.2) ; une fonction satisfaisant cette condition est d'ailleurs appelée fonction *localement lipschitzienne*.

est une fonction continue telle que, pour tout segment $[t_0, t_0+T]$ de I_0, pour tout $R>0$, il existe une constante $K_{[t_0,t_0+T],R}$ telle que

$$\begin{aligned} &\forall\, t \in [t_0, t_0+T], \quad \forall\, Y_1, Y_2 \in B_{\mathbb{R}^n}(0,R), \\ &\|F(t,Y_1) - F(t,Y_2)\| \leq K_{[t_0,t_0+T],R} \|Y_1 - Y_2\|\,, \end{aligned} \tag{6.4}$$

alors il existe, pour chaque segment $[t_0, t_0+T] \subset I_0$, pour chaque $Y_0 \in \mathbb{R}^n$, une unique fonction $t \in [t_0, t_0+T] \to \mathbb{R}^n$ de classe C^1 sur le segment $[t_0, t_0+T]$ (c'est-à-dire se prolongeant en une fonction de classe C^1 au voisinage de ce segment [2]) telle que

$$Y(t_0) = Y_0 \quad \& \quad \forall\, t \in [t_0, t_0+T],\ Y'(t) = F(t, Y(t)), \tag{6.5}$$

ce que l'on peut résumer en

$$\forall\, t \in [t_0, t_0+T], \quad Y(t) = Y_0 + \int_0^t F(\tau, Y(\tau))\, d\tau. \tag{6.6}$$

Lorsque le segment $[t_0, t_0+T] \subset I_0$ sera donné, nous nous intéresserons dans ce chapitre (Section 6.3), sous l'angle de l'algorithmique numérique, à la manière de calculer numériquement une solution approchée $t \in [t_0, t_0+T] \longmapsto Y_{\rm app}(t)$ à l'équation intégrale figurant en (6.6).

Exemple 6.2 (équations différentielles linéaires). Lorsque $U = I_0 \times \mathbb{R}^n$, où I_0 est un intervalle de $\mathbb{R}$ et que la fonction F est de la forme $F(t,Y) = A(t)\cdot Y + B(t)$, où A et B sont respectivement des applications continues de I_0 dans l'espace $\mathcal{M}_{n,n}$ des matrices réelles et de $\mathbb{R}$ dans $\mathbb{R}^n$, les conditions (6.6) sont vérifiées pour tout segment $[t_0, t_0+T]$ inclus dans I_0. Une telle équation différentielle $Y'(t) = A(t)\cdot Y(t) + B(t)$ est dite *linéaire*. La résolution d'une équation linéaire se traite *via* la *formule de Lagrange* par « double quadrature ». Si l'on note, pour $\tau \in I_0$ et $X \in \mathbb{R}^n$,

$$t \mapsto R(t,\tau)\cdot X$$

l'unique solution de l'équation homogène $Z'(t) = A(t)\cdot Z(t)$ telle que $Z(\tau) = X$ (la recherche de cette solution correspond à une première quadrature), la solution (I_0, Y) du problème de Cauchy (6.3) est donnée par

$$Y(t) = R(t,t_0)\cdot Y_0 + \int_{t_0}^t R(t,\tau)\cdot B(\tau)\, d\tau, \quad t \in I_0. \tag{6.7}$$

Si l'on sait ainsi résoudre l'équation différentielle homogène, c'est à dire « sans second membre », avec données initiales Y_0 arbitraires, on sait donc (en principe) résoudre grâce à la formule de Lagrange (6.7) l'équation différentielle $Y'(t) = A(t)\cdot Y(t) + B(t)$ avec donnée initiale $Y(t_0) = Y_0$. Il faut cependant avoir conscience que, même dans ce cas, à moins que $t \mapsto A(t)$ ne soit une fonction constante, on ne sait pas (en général) résoudre l'équation homogène $Z'(t) = A(t)\cdot Z(t)$ avec données initiales arbitraires $Z(\tau) = X$ autrement que numériquement ! Autrement dit, même dans le cas linéaire (pourtant relativement simple !), l'approche numérique est en général incontournable.

2. De fait, la fonction se prolonge en une fonction de classe C^1 à I_0 tout entier, cette fonction vérifiant d'ailleurs $Y'(t) = F(t, Y(t))$ pour tout $t \in I_0$.

Il faut noter que la résolution [3] des équations différentielles d'ordre supérieur

$$y^{(n)} = F(t, y, y', ..., y^{(n-1)}),$$

où F est une fonction continue dans un ouvert U de $\mathbb{R}^{n+1}$ vérifiant la condition (6.2) dans cet ouvert, se ramène à celle des équations différentielles $Y' = F(t, Y)$. Il suffit en effet de remarquer que dire que (I, y) vérifie l'équation d'ordre n (6.8) avec les conditions initiales (6.9) équivaut à dire que (I, Y), où $Y(t) = (y(t), y'(t), ..., y^{(n-1)}(t))$ vérifie le système

$$\begin{aligned}
&Y_0'(t) = Y_1(t)\\
&Y_1'(t) = Y_2(t)\\
&\vdots\\
&Y_k'(t) = Y_{k+1}(t)\\
&\vdots\\
&Y_{n-2}'(t) = Y_{n-1}(t)\\
&Y_{n-1}'(t) = F(t, Y_0(t), ..., Y_{n-1}(t)).
\end{aligned} \tag{6.10}$$

avec les conditions initiales $Y(t_0) = (y_{0,0}, ..., y_{0,n-1})$.

Remarque 6.3 (le cas linéaire à coefficients constants). Dans le cas particulier où I_0 est un intervalle contenant $[0, +\infty[$, la résolution des équations différentielles d'ordre supérieur

$$y^{(n)}(t) = a_0\, y(t) + a_1\, y'(t) + \cdots + a_{n-1}\, y^{(n-1)}(t) + b(t), \quad t \in I_0,$$

où $a_0, ..., a_{n-1}$ sont des constantes (réelles ou complexes) et b une fonction continue sur I_0 (avec données initiales prescrites $y_0, ..., y_{0,n-1}$ à l'origine pour les dérivées jusqu'à l'ordre $n-1$) se traite *via* le *calcul symbolique* sur lequel nous reviendrons. Ces équations sont très importantes en théorie de l'information où elles apparaissent sous forme discrétisées (équations aux différences impliquées dans le *filtrage digital*).

6.2. Quelques aspects qualitatifs

Soient U un ouvert de $\mathbb{R}^{n+1}$, F une fonction $(t, Y) \longmapsto F(t, Y)$ continue de U dans $\mathbb{R}$ et vérifiant la condition de Lipschitz (6.2). Nous sommes donc, en ce qui concerne l'équation différentielle $Y' = F(t, Y)$ (posée dans U), en situation de pouvoir appliquer le théorème de Cauchy-Lipschitz (Théorème 6.1). Nous allons ici décrire quelques aspects qualitatifs (et non plus quantitatifs) susceptibles d'aiguiller ultérieurement l'attaque numérique du problème telle qu'elle sera envisagée dans la Section 6.3 suivante. Nous nous intéresserons ici à deux situations :

– le cas $n = 1$;

3. On cherche, pour $(t_0, y_{0,0}, ..., y_{0,n-1}) = (t_0, Y_0) \in U$, les couples (I, y) tels que I soit un intervalle de $\mathbb{R}$ avec $(t, y(t), y'(t), ..., y^{(n-1)}(t)) \in U$ pour tout $t \in I$,

$$y^{(n)}(t) = F(t, y(t), y'(t), ..., y^{(n-1)}(t)) \quad \forall\, t \in I \tag{6.8}$$

et que soient remplies les conditions initiales

$$y(t_0) = y_{0,0},\ y'(t_0) = y_{0,1}, ..., y^{(n-1)}(t_0) = y_{0,n-1} \quad \text{(conditions initiales)}, \tag{6.9}$$

– le cas $n = 2$, mais où $U = I_0 \times \Omega$ et la fonction F ne dépend que de la variable $Y = (x, y) \in \Omega$, *i.e* $F(t, Y) = F(Y) = F(x, y)$ (de tels systèmes sont dits *autonomes*, cette notion pouvant d'ailleurs être étendue au cadre de la dimension $n > 2$)

6.2.1. Le cas $n = 1$. Nous supposons donc ici que U est un ouvert de $\mathbb{R}^2$ et que $(t, y) \mapsto f(t, y)$ est une fonction continue de U dans $\mathbb{R}$ telle que soit satisfaite la condition de Lipschitz (6.2).

EXEMPLE 6.4 (l'équation de Liouville). Un exemple modèle dans cette sous-section sera celui de l'*équation de Liouville*

$$y' = f(t, y) = t + y^2 \tag{6.11}$$

avec $U =]0, +\infty[\times\mathbb{R}$.

Les graphes des solutions (I, y) du probème de Cauchy (6.3) qui sont « maximales » au sens du second item du Théorème 6.1 sont appelées *trajectoires* ou *courbes intégrales* de l'équation différentielle $y'(t) = f(t, y(t))$. Le tracé de l'ensemble de ces courbes intégrales constitue le *plan de phase* de l'équation. Le théorème de Cauchy-Lipschitz nous permet de dessiner les courbes intégrales en nous aidant à les « piéger » dans des secteurs de l'ouvert $U \subset \mathbb{R}^2$ dans lequel la fonction f est définie. Cette capacité à « piéger » se concrétise par la mise en place de « barrières ».

DÉFINITION 6.5 (les diverses notions de « barrière »). On appelle *barrière inférieure forte* (resp. *faible*) pour l'équation différentielle $y' = f(t, y)$ tout couple (J, ψ) constitué d'un intervalle ouvert de $\mathbb{R}$, d'une fonction $\psi : J \longrightarrow \mathbb{R}$ de classe C^1 telle que

$$\forall\, t \in J\,,\ (t, \psi(t)) \in U \text{ et } \psi'(t) < f(t, \psi(t)) \text{ (resp. } \psi'(t) \leq f(t, \psi(t)) \text{)}\,. \tag{6.12}$$

On appelle *barrière supérieure forte* (resp. *faible*) pour l'équation différentielle $y' = f(t, y)$ tout couple (J, ψ) constitué d'un intervalle ouvert de $\mathbb{R}$, d'une fonction $\psi : J \longrightarrow \mathbb{R}$ de classe C^1 telle que

$$\forall\, t \in J\,,\ (t, \psi(t)) \in U \text{ et } \psi'(t) > f(t, \psi(t)) \text{ (resp. } \psi'(t) \geq f(t, \psi(t)) \text{)}\,. \tag{6.13}$$

Considérons une solution (I, y) et l'équation $y' = f(t, y)$, une barrière inférieure forte (J, ψ), et supposons qu'il existe $t_0 \in I \cap J$ tel que $y(t_0) = \psi(t_0)$. Nous allons montrer que

$$E = \{t \in]t_0, +\infty[\cap J \cap I\,;\ \psi(t) \geq y(t)\} = \emptyset\,.$$

Si tel n'était pas le cas, E aurait une borne inférieure t_1 ; la définition de E implique $t_0 < t_1$ et $y - \psi > 0$ sur $]t_0, t_1[$. De par la continuité de ψ et de y, on aurait $y(t_1) = \psi(t_1)$; mais $y'(t_1) = f(t_1, y(t_1)) > \psi'(t_1)$, ce qui montre que $y - \psi$ serait strictement croissante au passage de t_1, donc strictement négative avant t_1, contredisant ainsi la définition de t_1 comme borne inférieure de E. Ainsi, au-delà d'un instant t_0 où une barrière inférieure forte a été franchie par une courbe intégrale de l'équation, il devient impossible que cette même courbe intégrale la franchisse à nouveau à un instant $t > t_0$: on dit que la barrière est devenue *fortement infranchissable*. Le même raisonnement vaut pour les barrières supérieures fortes. En ce qui concerne les barrières inférieures ou supérieures *faibles*, une fois franchies par une courbe intégrale, elles peuvent ultérieurement être « touchées » une nouvelle fois par cette même courbe intégrale, mais en aucun cas traversées : on dit qu'elles sont devenues

faiblement infranchissables. Ces règles simples (elles reposent sur ce que l'on appelle communément le *principe de comparaison*) sont souvent combinées avec le lemme important suivant, complétant le théorème de Cauchy-Lipschitz, dit « lemme des bouts », que l'on peut énoncer de manière heuristique de la manière suivante :

LEMME 6.6 (lemme des « bouts »). *Sous les conditions d'application du Théorème de Cauchy-Lipschitz 6.1, les extrémités (ou encore les « bouts ») des courbes intégrales de l'équation différentielle* $y' = f(t, y)$ *sont toujours des points situés au bord de l'ouvert* U *dans lequel est définie, continue (et vérifie la condition de Lipschitz* (6.2)*) la fonction* f.

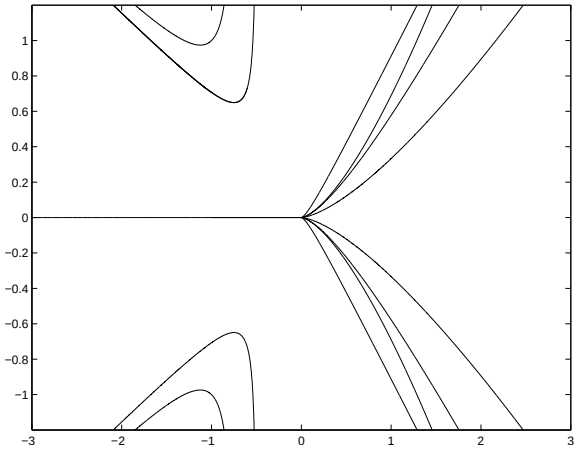

FIGURE 1. Plan de phase de l'équation de Bernoulli $y' = y/t + y^3/t^4$ si $U = \mathbb{R} \times \mathbb{R}$

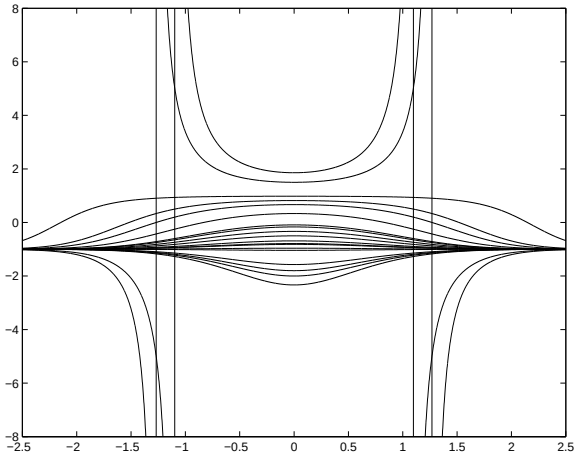

FIGURE 2. Plan de phase de l'équation à variables séparées $y' = x(y^2 - 1)$ dans $\mathbb{R} \times \mathbb{R}$

Les notions de barrière, faible ou forte, le lemme des bouts, *etc.*, sont autant d'outils permettant ainsi de localiser les trajectoires et d'esquisser le dessin du plan de phase, préparant ainsi le terrain pour une approche numérique telle que nous l'envisagerons dans la Section 6.3 suivante. Les courbes intégrales se retrouvent par exemple « piégées » entre deux barrières (inférieure et supéreure) devenant, une fois franchies, infranchissables ; ces situations sont dites situations d'« *entonnoir* ».

On s'aide pour ce tracé de la représentation préliminaires des courbes intégrales (I, y) correspondant aux solutions dites *stationnaires*, *i.e* telles que $t \mapsto y(t)$ reste constant sur leur intervalle de définition I, s'il en existe. La recherche des courbes stationnaires (trajectoires horizontales dans le plan de phase) revient à la recherche des nombres réels y_0 tels que la fonction $t \mapsto f(t, y_0)$ soit identiquement nulle sur un intervalle I tel que $(t, y_0) \in U$ pour tout $t \in I$. Sur les deux figures 1 et 2, on a ainsi représenté les plans de ohase de deux équations simples. Les solveurs numériques d'équations différentielles ordinaires (`ode` sous `MATLAB`, basés sur l'utilisation des méthodes numériques que nous présenterons dans la Section 6.3 suivante) permettent la représentation la plus exhaustive possible de plans de phase. Les exemples des équations de Bernoulli ou à variables séparées présentés sur les figures 1 et 2 ont été traités par intégration théorique (et non numérique) de l'équation différentielle, celle ci s'avérant dans ce cas très particulier possible.

EXEMPLE 6.7 (à nouveau l'exemple de l'équation de Liouville). Reprenons l'exemple de l'équation de Liouville (exemple 6.11) et notons (I, y) la solution de cette équation valant y_0 à l'instant $t_0 > 0$. Nous ne savons pas calculer cette fonction y, mais savons en revanche résoudre les équations à variables séparées

$$y' = \gamma^2 + y^2$$

lorsque $\gamma \in \mathbb{R}^*$. La solution maximale du problème de Cauchy

$$y' = \gamma^2 + y^2\,,\ y(t_0) = y_0$$

(posé dans $U =]0, +\infty[\times\mathbb{R}$) est le couple (I_γ, y_γ), où

$$I_\gamma = \Big]\max\Big(0, t_0 - \frac{\pi}{2\gamma}\Big)\,,\ t_0 + \frac{\pi}{2\gamma}\Big[\,,\ y_\gamma(t) = \gamma \arctan\left[\gamma(t - t_0)\right].$$

Pour $t \geq t_0$ et $0 < \gamma < \sqrt{t_0}$, on a

$$y'_\gamma(t) = \gamma^2 + y_\gamma^2(t) < t_0 + y_\gamma^2(t) \leq f(t, y_\gamma(t))\,,$$

ce qui fait de (I_γ, y_γ) une barrière inférieure forte, devenant infranchissable à partir de l'instant t_0 où elle a été franchie ($y(t_0) = y_\gamma(t_0) = y_0$). L'existence d'une telle barrière forte contraint notre solution y à n'avoir, au-delà de t_0, qu'une durée de vie ne pouvant excéder $t_0 + \frac{\pi}{2\gamma}$; comme γ est arbitraire pourvu que $\gamma < \sqrt{t_0}$, la borne supérieure de l'intervalle I vaut $t_0 + \frac{\pi}{2\sqrt{t_0}}$. Le lemme des bouts (Lemme 6.6) nous assure de plus que

$$\lim_{t \to t_0 + \frac{\pi}{2\sqrt{t_0}}} y(t) = +\infty\,.$$

Ainsi se trouve esquissé le tracé de notre courbe intégrale (I, y) à droite de t_0.

6.2.2. Le cas autonome lorsque $n = 2$. Les systèmes différentiels du type

$$\begin{aligned} \frac{dx}{dt} &= \Phi(x(t), y(t)) \\ \frac{dy}{dt} &= \Psi(x(t), y(t)) \end{aligned} \tag{6.14}$$

induisent des notions (pouvant sembler *a priori* différentes, bien que ce ne soit pas en fait le cas) de *trajectoire* et de *plan de phase*. Le plan de phase est cette fois la représentation dans le plan (plus précisément dans l'ouvert U de $\mathbb{R}^2$ où sont définies et continues les deux fonctions Φ et Ψ) des *trajectoires*, c'est-à dire ici des courbes paramétrées (ce ne sont plus des graphes comme dans le cas précédent)

$$t \in I \mapsto (x(t), y(t))$$

solutions du système différentiel autonome (6.14). Le théorème de Cauchy-Lipschitz impose que par chaque point (x_0, y_0) de U ne passe qu'une et une seule trajectoire. Une trajectoire ne saurait d'autre part admettre de point double. Cependant, certaines trajectoires peuvent être fermées (être des lacets sans points doubles), on les appelle des *orbites*. Les trajectoires réduites à un singleton sont dites *stationnaires* : si le point initial est ce singleton, alors on n'en bouge pas !

Exemple 6.8 (le modèle « proie-prédateur »). Le modèle *proie-prédateur* (ou de *Lotka-Volterra*[4]) est aujourd'hui un modèle de système autonome très classique en dynamique des populations et dans nombre de questions relevant de questions appliquées. Il permet d'introduire le concept important de *stabilité* (que nous retrouverons plus loin dans le contexte de la résolution numérique des EDO, voir la définition 6.16). Le modèle (continu) de ce système d'évolution est le suivant : a, b, c, d désignant quatre paramètres réels strictement positifs, il s'agit du système différentiel

$$\begin{aligned} x'(t) &= x(t)\,(a - by(t)) \\ y'(t) &= y(t)\,(-c + dy(t)). \end{aligned} \tag{6.15}$$

(ici $U = \mathbb{R} \times \mathbb{R}^2$). L'interprétation correspondant à ce modèle est la suivante : deux types de population cohabitent. La première (dont l'évolution est matérialisée en termes de proportion par x est l'effectif des *proies*) se développe exponentiellement en $\exp(at)$; la seconde (effectif des *prédateurs*, matérialisée en termes de proportion par y) s'éteint exponentiellement en $\exp(-ct)$; le facteur b s'interprète comme la *pression de prédation*, le facteur d comme l'*accessibilité des proies*. Ces modèles se retrouvent couramment en épidémiologie et, bien sûr, les questions de propagation de virus en sécurité informatique font qu'on les croise également en informatique et sécurité réseaux. Les modèles plus réalistes sont les modèles perturbés où l'on suppose que le taux de croissance x des proies diminue lorsque la population augmente (du fait de contraintes environnementales ou de subsistance par exemple). Le modèle du système d'évolution est alors

$$\begin{aligned} x'(t) &= x(t)\,(a - \epsilon x(t) - by(t)) \\ y'(t) &= y(t)\,(-c + dx(t)), \end{aligned} \tag{6.16}$$

4. Le mathématicien et statisticien autrichien Alfred James Lotka (1880-1949) et le mathématicien et physicien italien Vito Volterra (1860-1940) l'introduisirent vers 1925, ouvrant la voie à la dynamique des populations.

où ϵ est un cinquième paramètre. L'attaque de ce type de problème se fait numériquement par discrétisation (avec les méthodes que nous présenterons en dimension $n = 1$ dans la section suivante). Il faut noter cependant que le système (6.15) présente deux points stationnaires (trajectoires réduites à un point), à savoir $(0, 0)$ et $(c/d, a/b)$, de nature différente :

- si l'on perturbe l'origine en initiant la trajectoire en un point voisin (x_0, y_0), on constate que la nouvelle trajectoire initiée en (x_0, y_0) s'éloigne de l'origine (on dit que l'origine est un point d'équilibre *instable*) ;
- au contraire, si l'on perturbe le point $(c/d, a/b)$ en initiant la trajectoire en un point voisin, la nouvelle trajectoire reste une orbite autour du point $(c/d, a/b)$; on dit que $(c/d, a/b)$ est un point d'équilibre *stable*.

Les notions d'*instabilité* et de *stabilité* pour les équilibres sont fondamentales dans les questions relevant de l'analyse qualitative des systèmes différentiels autonomes (plus généralement des équations différentielles $Y'(t) = F(t, Y(t))$, autonomes ou non). Au voisinage d'un point d'équilibre stable, l'étude d'un système autonome de type (6.14) (comme par exemple (6.15)) peut être approchée par celle du système linéaire autonome obtenu en remplacant Φ et Ψ par leurs polynômes de Taylor à l'ordre 1 (donc des fonctions affines) au point d'équilibre (α, β). Ceci résulte d'un théorème majeur dans l'étude des systèmes dynamiques, le théorème de Lyapunov. C'est aussi par ce biais que l'on peut constater qu'un point d'équilibre (tel $(0, 0)$ pour le système (6.15)) est instable : les valeurs propres de la matrice $(2, 2)$ du système linéarisé sont ici deux nombres réels non nuls de signe opposés, ce qui correspond à une configuration de *point-selle* et donc à une situation d'équilibre instable (certaines trajectoires sont attirées, d'autres sont repoussées). Sans chercher à linéariser le problème au voisinage d'un des deux points d'équilibre, on pourra également envisager l'approche numérique à la résolution des systèmes autonomes (6.15) ou (6.16) en utilisant les schémas numériques décrits dans la Section 6.3 dans le cadre $n = 1$, mais qu'il est aisé de transporter au cas $n = 2$ (Euler explicite, Craig-Nicholson, Runge-Kutta, *etc.*).

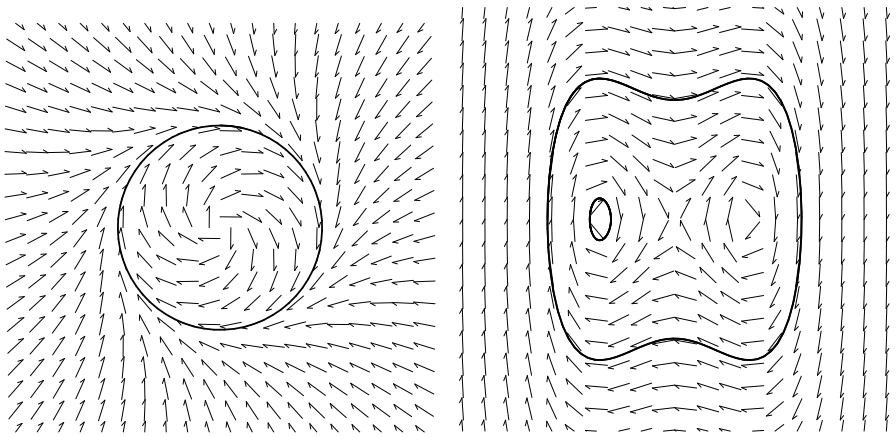

FIGURE 3. Deux systèmes « perturbés »

EXEMPLE 6.9 (le plan de phase de deux systèmes autonomes obtenus par perturbation d'une configuration stable). Un système linéaire avec l'origine comme

point d'équilibre stable (tel $x' = y, y' = -x$, où les trajectoires sont des cercles concentriques), lorsqu'il est « perturbé » en un système non linéaire, peut faire apparaitre des trajectoires spirales s'éloignant ou se rapprochant de la trajectoire « stationnaire » $Y = 0$ ou d'une trajectoire du système original. Par exemple (voir la figure de gauche sur la figure 3), les trajectoires du système $x' = y+x(1-x^2-y^2)$, $y' = -x+y(1-x^2-y^2)$ (ici $U = \mathbb{R}\times\mathbb{R}^2$) sont toutes « attirées » par le cercle unité, trajectoire particulière (sur laquelle la perturbation s'annule) du système originel $x' = y$, $y' = -x$. De même, les trajectoires du système $x' = y, y' = -2x(2x^2 - 1)$ (« perturbé » du système linéaire homogène $x' = y, y' = 2x$, pour lequel il y a attraction et rotation) ne sont plus des spirales, mais des courbes fermées dont l'allure dépend de la position des données initiales $Y(t_0)$ par rapport aux trois trajectoires stationnaires ($Y_0 = (0, 0)$, $Y_0 = (\pm 1/\sqrt{2}, 0)$), voir la figure de droite sur la figure 3. On pourra envisager l'approche numérique de ces problèmes en utilisant les schémas numériques décrits dans la section 6.3 dans le cadre $n = 1$, mais qu'il est aisé de transporter au cas $n = 2$ (Euler explicite, Craig-Nicholson, Runge-Kutta, *etc.*).

6.3. Résolution numérique des EDO

On se place dans le contexte présenté dans la section 6.1, où la condition de Lipschitz forte (6.4) est supposée remplie par la fonction F. On se limitera aussi ici au cas $n = 1$ (le cas général se traitant coordonnée-fonction par coordonnée-fonction ou matriciellement). On notera donc $F(t, y) = f(t, y)$ pour $(t, y) \in I_0 \times \mathbb{R}$.

6.3.1. Schémas numériques explicites ou implicites. Nous allons introduire ici un principe [5] basé sur la démarche suivante (lorsque $[t_0, t_0 + T] \subset I_0$) :

(1) on choisit un « pas maximal » $h_0 > 0$ (il faut, on le verra, être parfois soigneux, voir la Remarque 6.17 dans la section suivante) et une fonction continue
$$\Phi[f] \ : [t_0, t_0 + T] \times \mathbb{R} \times [0, h_0] \longrightarrow \mathbb{R}.$$
(2) pour $h = T/N \leq h_0$, on construit la suite récurrente $(y_{h,k})_{k\geq 0}$ solution de
$$\frac{y_{h,k+1} - y_{h,k}}{h} = \Phi[f](t_k, y_{h,k}, h), \quad k = 0, ..., N-1, \ (\text{ici } t_k = t_0 + k\,h) \tag{6.17}$$
et initiée à $y_{h,0} = y_0$, y_0 étant donné dans $\mathbb{R}$ (condition initiale).

L'objectif visé est que, si le pas h est fixé assez petit (en tout cas inférieur à h_0), $y_{h,k}$ approche la valeur de la solution f de l'équation différentielle $y'(t) = f(t, y(t))$ (avec condition initiale $y(t_0) = y_0$) au point $t_k = t_0 + k\,h$ (pour simplifier, on omet dans la notation t_k la dépendance implicite en h) du maillage
$$t_0 < t + h < t + 2h < \cdots < t_0 + (N-1)\,h < t_0 + Nh = t_0 + T.$$
On se souvient (Section 3.2) que
$$\frac{y_{h,k+1} - y_{h,k}}{h}$$
peut en effet être interprété comme la valeur approchée de $t \mapsto y'(t)$ soit au point médian de $[t_k, t_{k+1}]$ (calcul centré), soit au point t_k ou t_{k+1} (versions décentrées). Un tel schéma numérique est dit explicite car le calcul de $y_{h,k+1}$ se fait à partir de la connaissance de $y_{h,k}$ tant que $k = 0, ..., N-1$.

5. C'est le principe des méthodes dites, on verra plus loin pourquoi, « à un pas ».

On peut aussi envisager les *schémas implicites* où, dans l'étape 2 du processus décrit, on remplace (6.17) par

$$\frac{y_{h,k} - y_{h,k-1}}{h} = \Psi[f](t_k, y_{h,k-1}, y_{h,k}, h), \quad k = 1, ..., N, \text{ (ici } t_k = t_0 + k\,h), \tag{6.18}$$

où

$$\Psi[f] \ : (t, y, \xi, h) \in [t_0, t_0 + T] \times \mathbb{R} \times \mathbb{R} \times [0, h_0] \longmapsto \Psi(t, y, \xi, h) \in \mathbb{R}$$

est une fonction continue (toujours en maintenant la condition initiale $y_{h,0} = 0$) ; cette fois la détermination de $y_{h,k}$ à partir de $y_{h,k-1}$ passe par la résolution de l'équation implicite (d'inconnue ξ)

$$\frac{\xi - y_{h,k-1}}{h} = \Phi[f](t_k, y_{h,k-1}, \xi, h).$$

Qu'il s'agisse de la méthode explicite (basée sur (6.17)) ou implicite (basée sur (6.18)), la récurrence permettant de calculer de manière inductive les $y_{h,k}$ lorsque le pas $h \leq h_0$ est fixé est une récurrence à un terme ($y_{h,k+1}$ fonction de $y_{h,k}$). C'est la raison pour laquelle on appelle ces méthodes *méthodes à un pas*.

EXEMPLE 6.10 (les méthodes d'Euler explicite, implicite et modifiée). Si l'on prend

$$\Phi[f](t, y, h) = f(t, y)$$

dans (6.17) (cette fonction est indépendante de h dans ce cas), on obtient le schéma numérique dit *schéma d'Euler explicite*, bien connu depuis le lycée [6]. En prenant

$$\Psi[f]\,(t, y, \xi, h) = f(t, \xi)$$

dans (6.18), on obtient avec (6.18) le *schéma d'Euler implicite* (ou *rétrograde*). Si l'on souhaite respecter le fait que l'erreur dans le calcul numérique de dérivée est meilleure dans la situation centrée (voir la Section (3.2)), on choisit comme fonction $\Phi[f]$ la fonction

$$(t, y, h) \longmapsto f(t + h/2, y + h/2\, f(t, y)).$$

Cette fois la fonction Φ fait intervenir la variable h et le schéma numérique ainsi construit est le *schéma d'Euler explicite modifié*.

Une alternative pour remplacer les relations (6.17) ou (6.18), si l'on a en mémoire l'équivalence entre (6.5) et (6.6) pour traduire que $t \mapsto y(t)$ est solution du problème de Cauchy, est de penser à la résolution de l'équation sous la forme de la recherche d'une solution à l'équation intégrale

$$y(t) = y_0 + \int_0^t f(\tau, y(\tau))\, d\tau,$$

soit au jeu d'équations intégrales

$$y_{h,k+1} - y_{h,k} \simeq y(t_{k+1}) - y(t_k) = \int_{t_k}^{t_{k+1}} f(\tau, y(\tau))\, d\tau,$$

le membre de droite étant exprimé à partir d'une méthode numérique du type Newton-Cotes (voir la Section 5.2.1). Avant de voir ultérieurement (avec les méthodes dites de *Runge-Kutta*, que nous introduirons dans la section 6.3.3) des outils

6. On doit ce schéma numérique au mathématicien suisse Leonhard Euler (1707-1783), pionnier des mathématiques actuelles au siècle des lumières, qui l'introduisit dès 1768 sans doute.

issus de l'intégration numérique plus élaborés (méthodes de Simpson, de quadrature, *etc.*), nous donnons ici deux exemples de construction de schémas numériques implicites ou explicites à partir des formules à un point (rectangle) ou deux points (trapèzes).

EXEMPLE 6.11 (avec la méthode des rectangles : schéma d'*Euler modifié*). La formule à un point (rectangles) conduit par exemple à

$$y_{h,k+1} - y_{h,k} \simeq h\, f\Big(t_k + \frac{h}{2}\,,\, y\big(t_k + \frac{h}{2}\big)\Big).$$

En combinant avec

$$y\big(t_k + \frac{h}{2}\big) = y(t_k) + \frac{h}{2}\, y'(t_k) + o(h) = y(t_k) + \frac{h}{2}\, f\big(t_k, y(t_k)\big) + o(h),$$

on voit que le choix de $\Phi[f]$ est alors

$$\Phi[f] \ : (t, y, h) \longmapsto f\Big(t + \frac{h}{2}\,,\, y + \frac{h}{2} f(t, y)\Big). \tag{6.19}$$

On retrouve le *schéma d'Euler modifié*.

EXEMPLE 6.12 (avec la méthode des trapèzes : schéma explicite d'*Heun* et implicite de *Craig-Nicholson*). La formule à 2 points (trapèzes) conduit, elle, à

$$y_{h,k+1} - y_{h,k} \simeq \frac{h}{2}\Big(f\big(t_k, y(t_k)\big) + f\big(t_{k+1}, y(t_{k+1})\big)\Big). \tag{6.20}$$

En combinant avec

$$\begin{aligned} y(t_{k+1}) &= y(t_k + h) = y(t_k) + h\, y'(t_k) + o(h) \\ &= y(t_k) + f(t_k, y(t_k))\, h + o(h), \end{aligned} \tag{6.21}$$

on voit que le choix de $\Phi[f]$ dans (6.17) qui est adapté à cette méthode des trapèzes est

$$\Phi[f] \ : (t, y, h) \longmapsto \frac{1}{2}\Big(f(t, y) + f\big(t + h, y + h f(t, y)\big)\Big). \tag{6.22}$$

C'est le *schéma de Heun*[7] *explicite*. Mais on peut également ne pas utiliser l'approximation (6.21) et utiliser directement (6.20) pour générer le schéma (implicite cette fois)

$$\frac{y_{h,k} - y_{h,k-1}}{h} = \frac{1}{2}\Big(f(t_k, y_{h,k-1}) + f(t_k + h, y_{h,k})\Big) \tag{6.23}$$

On pose alors

$$\Psi \ : (t, y, \xi, h) \longmapsto \frac{1}{2}\Big(f(t, y) + f(t + h, \xi)\Big)$$

pour obtenir le schéma implicite du type (6.18) dit *schéma de Craig-Nicholson implicite*[8].

---

7. Du nom du mathématicien allemand Karl Heun (1859-1929) qui l'introduisit.

8. On doit cette méthode (initialement introduite pour la résolution de l'équation de la chaleur) à la mathématicienne britannique Phillis Nicholson (1917-1968) et à son compatriote le physicien John Craig (1916-2006).

6.3.2. Consistance, stabilité, convergence, ordre, d'une méthode à un pas. On se place ici dans le contexte de la section précédente, à savoir celui d'une équation différentielle $y'(t) = f(t, y(t))$, où f est une fonction continue de $I_0 \times \mathbb{R}$ dans $\mathbb{R}$, satisfaisant la condition de Lipschitz (6.4). On se donne un segment temporel $[t_0, t_0 + T]$ de I_0 et une donnée initiale $y_0 \in \mathbb{R}$ et l'on introduit l'un des schémas numériques (6.17) (explicite) ou (6.18) (implicite) conditionnés au choix d'une fonctionnelle $\Phi[f]$ ou $\Psi[f]$ (voir la section précédente).

Faisons une première observation qui va nous permettre d'unifier les deux situations (6.17) et (6.18). Du fait que dans le schéma numérique implicite (6.18), on puisse exprime $y_{h,k}$ de manière implicite en fonction de $y_{h,k-1}$, on peut reporter cette expression implicite dans

$$\Psi[f](t_k, y_{h,k-1}, y_{h,k}, h),$$

ce qui permet de transformer cette expression en une autre, certainement autrement plus complexe, mais de la forme

$$\Phi[f](t_k, y_{h,k-1}, h).$$

Cette remarque préliminaire nous autorise donc à traiter de la même manière schémas numériques implicites et schémas numériques explicites, ce que l'on fera dorénavant en ne considérant plus que des schémas numériques du type (6.17).

On note $y(t_k)$, $k = 0, ..., N$ ($t_k = t_0 + kh$), les valeurs aux points du maillage de pas $h = T/N$ de l'unique solution y (valant y_0 à l'instant t_0) de l'équation $y'(t) = f(t, y(t))$ sur l'intervalle $[t_0, t_0 + T]$. On note également, pour $k = 1, ..., N$, $y_{h,k}$ le terme général de la suite générée par le schéma numérique (6.17) initié à la même valeur $y_0 \in \mathbb{R}$.

On définit alors une « erreur » au pas $k - 1$ (ou à l'instant t_k) comme

$$e_k(h; y_0) := \frac{y(t_k) - y(t_{k-1})}{h} - \Phi[f]\,(t_{k-1}, y_{h,k-1}, h) \tag{6.24}$$

Cette erreur $e_k(h; y_0)$ peut être interprétée comme l'erreur que l'on commet au pas $k - 1$ en remplaçant ce qui devrait être la relation vérifiée par la solution exacte $t \mapsto y(t)$ du problème, soit l'équation intégrale

$$\frac{y(t_k) - y(t_{k-1})}{h} = \frac{1}{h} \int_{t_{k-1}}^{t_k} f(\tau, y(\tau))\, d\tau\,,$$

par la relation que vérifie la solution « approchée » interpolant les valeurs $y_{h,k}$ générées par le schéma numérique (6.17). L'erreur $e_k(h; y_0)$, $k = 1, ..., N$ en (6.24) est appelée *erreur de troncature* à l'instant $t_k = t_0 + kh$.

Définition 6.13 (consistance d'une méthode à un pas). Le schéma numérique (6.17) est dit *consistant* (par rapport à l'équation différentielle pour la résolution numérique duquel il a été conçu, à savoir $y'(t) = f(t, y(t))$ avec donnée initiale y_0 en t_0 précisée), si

$$\lim_{N \to +\infty} \max_{1 \le k \le N} |e_k(T/N; y_0)| = 0,$$

ce quelque soit la valeur initiale y_0.

Pour qu'un schéma numérique du type (6.17) soit consistant par rapport à l'équation différentielle $y'(t) = f(t, y(t))$, il faut nécessairement que, pour tout instant t de $[t_0, t_0 + T]$, pour tout $y \in \mathbb{R}$, on ait

$$\Phi[f](t, y, 0) = f(t, y). \tag{6.25}$$

Cette condition (6.25) est d'ailleurs en fait une condition nécessaire et suffisante de consistance du schéma numérique (6.17) avec l'équation différentielle sous-jacente $y'(t) = f(t, y(t))$.

Si une méthode à un pas (du type (6.17) est consistante par rapport à l'équation différentielle $y't) = f(t, y(t))$, on peut définir l'*ordre* de cette méthode à un pas. On se place pour simplifier ici sous l'hypothèse que la fonction f est de classe C^∞ sur I_0 ; ceci implique que toute solution y de classe C^1 de l'équation différentielle $y'(t) = f(t, y(t))$ sur I_0 est automatiquement C^∞ (d'après la règle de Leibniz). On fera constamment cette hypothèse par la suite.

Définition 6.14 (ordre d'une méthode à un pas). La méthode à un pas fondée sur le schéma numérique (6.17) est dite d'ordre au moins $p \in \mathbb{N}^*$ s'il existe une constante K_{y_0} (ne dépendant que de $\Phi[f]$, de h_0, et de la constante initiale y_0) telle que

$$\max_{1 \le k \le N} |e_k(h; y_0)| \le K_{y_0}\, h^p$$

pour $h \in [0, h_0]$. L'ordre est égal à p si p est le plus grand entier vérifiant cette propriété.

Exemple 6.15 (ordre des méthodes d'Euler, de Craig-Nicholson, de Heun). Si l'on se souvient (voir la section 5.2.3) que l'erreur dans la méthode des rectangles ou des trapèzes est en $O(h^3)$, et que l'on fait l'observation que la définition de l'erreur de troncature implique une division par h de ce qui devrait être

$$\int_{t_k}^{t_{k+1}} f(\tau, y(\tau))\, d\tau$$

(approchée par une méthode de Newton-Cotes à un ou deux points comme dans la section précédente), on observe que les méthodes d'Euler modifée (fondées sur la méthode des rectangles), de Craig-Nicholson ou Heun (fondées sur la méthodes des trapèzes) sont d'ordre $3 - 1 = 2$. En revanche, les méthodes d'Euler ou Euler rétrograde sont d'ordre 1.

Définissons à partir de la fonction $(t, y) \mapsto f(t, y)$, une suite de fonctions

$$(t, y) \mapsto f^{[l]}(t, y), \quad l = 0, 1, 2, ...$$

suivant la relation inductive

$$f^{[l+1]}(t, y) = \frac{\partial f^{[l]}}{\partial t}(t, y) + \frac{\partial f^{[l]}}{\partial y}(t, y)\, f(t, y), \tag{6.26}$$

ce à partir de $f^{[0]}(t, y) = f(t, y)$. Ainsi, si $t \mapsto y(t)$ est solution de $y'(t) = f(t, y(t))$, on a (d'après la règle de dérivation des fonctions composées $t \mapsto \varphi(u(t), v(t))$

$$f^{[l]}(t, y(t)) = y^{(l+1)}(t) = \frac{d^l}{dt^l}\,[f(t, y(t))].$$

Ceci étant posé, il est facile de dégager une condition nécessaire et suffisante pour qu'une méthode à un pas basée sur le schéma numérique explicite (6.17) soit d'ordre au moins p.

PROPOSITION 6.1. *La méthode à un pas basée sur le schéma numérique explicite* (6.17) *est d'ordre au moins p si et seulement si, pour tout couple (t, y) dans le domaine $[t_0, t_0 + T] \times \mathbb{R}$, on a les relations*

$$\frac{\partial^l \Phi[f]}{\partial h^l}(t, y, 0) = \frac{f^{[l]}(t, y)}{l+1}\,,\ l = 0, ..., p-1. \tag{6.27}$$

DÉMONSTRATION. Montrons d'abord que la condition est suffisante. D'après la formule de Taylor-Lagrange (Proposition 3.2), si y est une solution de l'équation différentielle $y'(t) = f(t, y(t))$, on a

$$\begin{aligned} y(t_{k+1}) - y(t_k) &= h\, y'(t_k) + \cdots + \frac{h^p}{p!} y^{(p)}(t_k) + \frac{h^{p+1}}{(p+1)!} y^{(p+1)}(\xi_{h,k}) \\ &\qquad (\xi_{h,k} \in [t_k, t_{k+1}],\ k = 0, ..., N-1) \\ &= \sum_{l=0}^{p-1} h^{l+1} \frac{f^{[l]}(t_k, y(t_k))}{(l+1)!} + h^{p+1} \frac{f^{[p]}(\xi_{h,k}, y(\xi_{h,k}))}{(p+1)!}. \end{aligned} \tag{6.28}$$

On écrit de même le développement en série de Taylor-Lagrange de

$$h \mapsto \Phi[f](t_k, y(t_k), h)$$

à l'ordre p en $h = 0$, ce qui donne

$$\begin{aligned} \Phi[f](t_k, y(t_k), h) &= \sum_{l=0}^{p-1} \frac{\partial^l [\Phi[f]]}{\partial h^l}(t_k, y(t_k), 0)\, \frac{h^l}{l!} + \frac{h^p}{p!} \frac{\partial^l [\Phi[f]]}{\partial h^l}(t_k, y(t_k), \eta_{h,k}), \\ &\qquad (\eta_{h,k} \in [0, h]). \end{aligned} \tag{6.29}$$

En soustrayant (6.29) à (6.28) (préalablement divisée par h), on trouve, pour tout entier $k = 1, ..., N$, une expression de $e_k(h; y_0)$. Pour que cette erreur s'estime en $O(h^p)$, on doit écrire précisément la nullité de tous les coefficients affectant les puissances de h d'exposant strictement inférieur à p. Ceci correspond justement aux conditions (6.27). □

La seconde notion importante pour un schéma numérique de type (6.17) est une notion qui ne concerne que le schéma numérique lui-même, sans qu'il ne soit fait référence à l'équation différentielle $y'(t) = f(t, y(t))$ pour lequel ce schéma est envisagé comme solveur numérique (autrement que dans l'expression bien sûr de la fonctionnelle $\Phi[f]$). C'est la notion de *stabilité*[9].

DÉFINITION 6.16 (stabilité d'un schéma numérique du type (6.17)). Le schéma numérique (6.17) est dit *stable* s'il existe des constantes absolues M_1 et M_2 (indépendantes de $h = T/N$) telles que, pour toute paire $(y_0, y_{0,\mathrm{pert}})$ de valeurs initiales, pour tout vecteur $(\epsilon_1, ..., \epsilon_N) \in \mathbb{R}^n$ (pensé comme petite « perturbation » du schéma

9. La notion de stabilité a déjà été évoquée dans ce cours à propos des aspects quantitatifs de l'étude des systèmes différentiels autonomes, sur le modèle « proie-prédateur », voir la section 6.2.2.

numérique), on a, si les $y_{h,k}$, $k = 0, ..., N$, sont générés par le schéma (6.17) et si les $y_{h,k,\text{pert}}$, $k = 0, ..., N$, sont générés par le schéma « perturbé »

$$\frac{y_{h,k+1,\text{pert}} - y_{h,k,\text{pert}}}{h} = \Phi[f](t_k, y_{h,k,\text{pert}}, h) + \epsilon_{k+1} \quad (k = 0, ..., N-1) \tag{6.30}$$

initié à $y_{h,0,\text{pert}} = y_{0,\text{pert}}$,

$$\max_{1,...,N} |y_{h,k,\text{pert}} - y_{h,k}| \leq M_1 |y_0 - y_{0,\text{perp}}| + M_2 \max_{k=1,...,N} |\epsilon_k|. \tag{6.31}$$

Si la fonction $\Phi[f]$ vérifie une condition de Lipschitz uniforme sur $[t_0, t_0 + T]$ (l'uniformité faissant référence ici à la dépendance en les variables y et h), du type

$$\begin{aligned} &\forall\, t \in [t_0, t_0 + T],\ \forall\, y, z \in \mathbb{R},\ \forall\, h \in [0, h_0], \\ &|\Phi[f](t, y, h) - \Phi[f](t, z, h)| \leq L_T\, |y - z|, \end{aligned} \tag{6.32}$$

où L_T est une constante positive, il est aisé de montrer que le schéma numérique correspondant (6.17) est stable. Cela résulte du fait suivant (facile à établir par récurrence et constituant la version discrète de ce que l'on appelle l'inégalité de Grönwall [10]) : si $(\eta_l)_{l\geq 0}$ et $(\epsilon_1)_{l\geq 0}$ sont deux suites de nombres positifs telles que

$$\eta_l \leq (1 + hL_T)\eta_{l-1} + \epsilon_l \quad \forall\, l \geq 1,$$

avec $h > 0$ alors

$$\eta_k \leq e^{L_T kh}\eta_0 + \sum_{l=1}^{k} e^{L_T(k-l)h}\epsilon_l \quad \forall\, k \geq 0 \tag{6.33}$$

(ici $t_k = t_0 + hk$).

Remarque 6.17 (la notion de « problème raide »). Comme cela apparait dans l'inégalité de Grönwall (forme discrète (6.33)), le fait que la constante $L_T \times T$ (L_T définie par la condition (6.32)) soit grande (et par voie de conséquence $\exp(L_T\, T)$ exponentiellement fois plus grande) se révèle un sérieux handicap pour le contrôle raisonnable des constantes M_1 et M_2 gouvernant la clause (6.31). On parle, pour de telles équations différentielles sur $[t_0, t_0 + T]$ (lorsque ce contrôle de $L_T \times T$ s'avère mauvais, soit parce que L_T est trop grande, soit parce que la durée de vie T est trop longue) de *problèmes raides*. De très strictes restrictions doivent alors être imposées au choix de h_0. Pour plus de détails ici (faute de temps, cette question n'a pu être traitée dans le cours), nous renvoyons au chapitre 3 de [**MathAp**], section IV.

Le dernier concept à préciser concernant un schéma numérique du type (6.17) (conçu comme solveur à un pas d'une équation différentielle $y'(t) = f(t, y(t))$) est celui de *convergence*.

Définition 6.18. Le schéma numérique à un pas (6.17), conçu comme solveur à un pas de l'équation différentielle $y'(t) = f(t, y(t))$ sur $[t_0, t_0 + T]$, est dit *convergent* (par rapport à cette équation différentielle) si et seulement si, pour tout y_0 dans $\mathbb{R}$, pour toute suite de points $y_{h,0}$ tendant vers y_0 lorsque $h = T/N$ dans vers 0, on a

$$\lim_{N\to+\infty} \max_{k=1,...,N} |y_{h,k} - y(t_k)| = 0, \tag{6.34}$$

10. Thomas Håkon Grönwall (1877-1932), physicien et mathématicien suédois à qui l'on doit (en 1919) cette inégalité importante, utilisée ici dans le cadre discret.

où $(y_{h,k})_{k=1,\dots,N}$ désigne la suite générée par le schéma numérique (6.17) à partir de la donnée initiale $y_{h,0}$, et où la fonction $t \mapsto y(t)$ désigne la solution de l'équation différentielle $y'(t) = f(t, y(t))$ valant précisément y_0 en $t = t_0$.

La règle importante suivante est à retenir :

PROPOSITION 6.2 (condition suffisante pour la convergence d'un solveur à un pas d'EDO). *Consistance et stabilité impliquent la convergence d'un solveur à un pas d'une équation différentielle ordinaire $y'(t) = f(t, y(t))$ sur un segment temporel $[t_0, t_0 + T] \subset I_0$ donné (la condition (6.4) étant remplie par f sur $I_0 \times \mathbb{R}$).*

6.3.3. Les méthodes de Runge-Kutta. On pourrait évidement pour réaliser un schéma numérique d'ordre au moins égal à p pour une équation différentielle $y'(t) = f(t, y(t))$ donnée (f étant supposée C^∞ sur $I_0 \times \mathbb{R}$) se baser sur la Proposition 6.1 et proposer comme fonction $\Phi[f]$ la fonction

$$(t, y, h) \longmapsto \sum_{l=0}^{p-1} \frac{h^l}{l+1} f^{[l]}(t, y).$$

Sous l'hypothèse (6.32), le schéma numérique associé à ce choix de $\Phi[f)$ est stable. D'après la Proposition 6.2, ce schéma est donc convergent relativement à l'EDO pour laquelle il est conçu. Cependant le caractère analytique souvent très complexe des expressions à évaluer $f^{[l]}(t, y)$ nous fait préférer à cette construction un choix algorithmiquement plus judicieux, basé une fois encore sur l'idée d'intégration approchée (Newton-Cotes ou quadrature). La démarche que nous allons décrire soutend la construction des schémas numériques de Runge-Kutta [11] .

La méthode est basé sur le calcul approché de

$$\int_{t_k}^{t_{k+1}} f(\tau, y(\tau))\, d\tau$$

figurant dans l'équation intégrale

$$y(t_{k+1}) - y(t_k) = \int_{t_k}^{t_{k+1}} f(\tau, y(\tau))\, d\tau$$

par une « double quadrature » (c'est-à-dire une double intégration).

La première quadrature est attachée au choix d'une suite de points (ordonnée de manière croissante, les points pouvant fort bien être répétés, ce que l'on a intérêt d'ailleurs à faire) $\xi_1, ..., \xi_K$ de $[0, 1]$. Cette suite correspond donc à un « maillage » (avec répétitions éventuelles) du segment $[0, 1]$. Ce maillage induit un maillage de chaque segment $[t_k, t_{k+1}]$ (comme toujours $t_k = t_0 + kh$) du type

$$t_k \leq t_k + \xi_1 h = t_{k,1} \leq t_k + \xi_2 h = t_{k,2} \leq \cdots \leq t_k + \xi_K h = t_{k,K} \leq t_{k+1}.$$

À chaque point ξ_j, $j = 1, ..., K$, on associe un « poids » $\beta_j \in \mathbb{R}$ de manière à ce que

$$\int_0^1 \varphi(u)\, du \simeq \sum_{j=1}^{K} \beta_j\, \varphi(\xi_j) \tag{6.35}$$

11. Cette démarche a été introduite par Carl Runge, mathématicien et physicien allemand (1856-1927) et développée numériquement par le mathématicien allemand Martin Kutta (1867-1944), connu aussi pour ses travaux en aérodynamique.

exacte au moins jusqu'à l'ordre 0 pour toute fonction φ continue sur $[0,1]$, ce qui signifie

$$\sum_{j=1}^{K} \beta_j = 1. \tag{6.36}$$

EXEMPLE 6.19 (Runge-Kutta « classique » : les données). Le choix de $\xi_1 = 0$, $\xi_2 = 1/2$, $\xi_3 = 1/2$, $\xi_4 = 1$ (ξ_2 étant pensé comme $1/2^-$ et ξ_2 comme $1/2^+$) et de $\beta_1 = 1/6$, $\beta_2 = \beta_3 = 1/3$, $\beta_4 = 1/6$, en adéquation avec la formule de Simpson ((5.21), section 5.2.1), sera un choix « modèle » pour nous car il conduira à la construction d'un solveur d'ordre $4 = 5 - 1$ (on se souvient que l'erreur dans la méthode de Simpson est en $O(h^5)$, voir la section 5.2.3).

En utilisant le changement de variables $t = t_k + uh$, on voit que la valeur approchée de

$$\int_{t_k}^{t_k+1} \psi(t)\, dt = h \int_0^1 \psi(t_k + uh)\, du,$$

via la formule de quadrature approchée (6.35) fournit l'approximation

$$\int_{t_k}^{t_{k+1}} f(\tau, y(\tau))\, d\tau = h \sum_{j=1}^{K} \beta_j\, f(t_{k,j}, y(t_{k,j})). \tag{6.37}$$

Pour définir la seconde quadrature qui va nous permettre de calculer des valeurs approchées $y_{k,j}$, $j = 1, .., K$, à la place des valeurs exactes $y(t_{k,j})$ de la solution de l'équation différientielle (valant y_0 à l'instant t_0) aux nœuds intermédiaires $t_{k,j}$, $j = 1, ..., K$, nous introduisons un tableau $K \times K$ de nombres réels $a_{j,l}$, $j = 1, ..., K$ (indice de ligne), $l = 1, ..., K$ (indice de colonne). Pour chaque $j = 1, ..., K$, le vecteur ligne $(a_{j,1}, ..., a_{j,K})$ doit ête pensé comme le vecteur des coefficients impliqués dans la formule de quadrature suivante [12] :

$$\int_0^{\xi_j} \varphi(u)\, du \simeq \sum_{l=1}^{K} a_{j,l}\, \varphi(\xi_j). \tag{6.38}$$

Cette formule de quadrature induit par changement de variables les formules approchées de quadrature suivantes :

$$\int_0^{t_{k,j}} \psi(t)\, dt \simeq h \sum_{l=1}^{K} a_{j,l}\, \psi(t_{j,l}). \tag{6.39}$$

Ces formules approchées (6.39) permettent de calculer des valeurs approchées $y_{k,j}$ à partir des approximations

$$\begin{aligned} y(t_k) + \int_0^{t_{k,j}} f(\tau, y(\tau))\, d\tau &\simeq y_k + h \sum_{l=1}^{K} a_{j,l} f(t_{j,l}, y(t_{k,l})) \\ &\simeq y_k + h \sum_{l=1}^{K} a_{j,l} f(t_{j,l}, y_{k,l}) = y_{k,j} \\ &\quad (j = 1, ..., K). \end{aligned} \tag{6.40}$$

12. Il est bien sûr naturel de choisir les a_{jl} nuls si $\xi_l > \xi_j$ mais il ne faut pas oublier qu'il y a des répétitions dans la suite des ξ_j, ce qui empêche d'affirmer que la matrice des $a_{j,l}$ soit forcément triangulaire inférieure !

Nous avons là un jeu d'équations en général implicite (il est explicite uniquement lorsque la matrice des $a_{j,l}$ a tous ses coefficients nuls sur la diagonale et au dessus) permettant de déterminer les $y_{k,j}$, $j = 1, ..., K$ à partir de y_k. On admet ici que si f vérifie la condition (6.32) et si

$$h_0 < \frac{1}{L_T \times (\text{rayon spectral}) ([a_{j,l}])}, \tag{6.41}$$

le théorème du point fixe assure la possibilité de résoudre implicitement le jeu d'équations (6.40) en les $y_{k,j}$, $j = 1, ..., K$, à partir de la connaissance de y_k (lorsque $h \leq h_0$) . Ceci est évidemment immédiat à faire si $a_{j,l} = 0$ pour tout $l \geq j$ (auquel cas la résolution est explicite) ; c'est plus compliqué dans le cas général, plus facile toutefois si la matrice $[a_{j,l}]_{j,l}$ est triangulaire inférieure (on dit que le schéma est *semi-explicite* dans ce cas).

Une fois résolu le jeu d'équations (implicites ou explicites) permettant de calculer à partir de y_k et des relations (6.40) les valeurs $y_{k,1}$, ..., $y_{k,K}$, on exprime y_{k+1} à partir de ces valeurs en utilisant la relation

$$\frac{y_{k+1} - y_k}{h} = \sum_{j=1}^{K} \beta_j \, f(t_{k,j}, y(t_{k,j})) \simeq \sum_{j=1}^{K} \beta_j \, f(t_{k,j}, y_{k,j}).$$

EXEMPLE 6.20 (l'exemple de Runge-Kutta « classique »). On reprend les données de l'exemple 6.19 ; il s'agit d'un exemple où la matrice $[a_{j,l}]_{j,l}$ (ici 4×4) a tous ses termes nuls sur la diagonale et au dessus. La méthode est donc ici explicite et le calcul de $y_{k,1}, ..., y_{k,4}$ peut être conduit explicitement de proche en proche. Ce calcul explicite menant de y_k à y_{k+1} peut se conduire ainsi (en introduisant le calcul de quatre constantes intermédiaires $A_{k,1}$, $A_{k,2}$, $A_{k,3}$ et A_{k_4} suivant ce scénario très facilement implémentable :

(6.42)

$$\begin{aligned}
&A_{k,1} := f(t_{k,1}, y_k) = f(t_k, y_k) \quad \text{car} \quad t_{k,1} = t_k \\
&A_{k,2} := f(t_{k,2}, y_k + h\, A_{k,1}/2) = f(t_k + h/2, y_k + h\, A_{k,1}/2) \quad \text{car} \quad t_{k,2} = t_k + h/2 \\
&A_{k,3} := f(t_{k,3}, y_k + h\, A_{k,2}/2) = f(t_k + h/2, y_k + h\, A_{k,2}/2) \quad \text{car} \quad t_{k,3} = t_k + h/2 \\
&A_{k,4} := f(t_{k,3}, y_k + h\, A_{k,3}) \quad \text{car} \quad t_{k,4} = t_k + h = t_{k+1} \\
&y_{k+1} := y_k + \frac{h}{6}\big(A_{k,1} + 2A_{k,2} + 2A_{k,3} + A_{k,4}\big).
\end{aligned}$$

La méthode ainsi construite, directement issue de la méthode de Newton-Cotes à trois points (Simpson) dont on sait qu'elle induit une erreur en $O(h^5)$ (voir la section 5.2.3), est d'ordre $4 = 5 - 1$ (il y a une division par h une fois approchée l'intégrale). Ce solveur explicite d'ordre 4 est très utilisé dans la pratique : c'est la méthode de Runge-Kutta dite « classique ». Cette méthode est consistante et stable, puisque le rayon spectral de A est nul (la matrice A est nilpotente dans ce cas).

On peut en utilisant la Proposition 6.1 trouver une condition néssaire et suffisante portant sur la matrice $[a_{j,l}]_{j,l}$ et le choix du vecteur ligne $(\beta_1, ..., \beta_K)$ pour que la méthode soit d'ordre 2. Outre la condition (6.36) qui assure juste la consistance, on doit imposer, pour que la méthode soit d'ordre 2, les deux conditions algébriques

supplémentaires

$$ {}^t\beta \cdot A \cdot \mathtt{ones}\,[1,K] = {}^t\beta \cdot \mathrm{diag}\,[\xi_1,..,\xi_K] \cdot \mathtt{ones}\,[1,K] = \frac{1}{2}. \tag{6.43} $$

Il suffit pour voir cela d'écrire le jeu de relations

$$ \begin{aligned} y_{\mathrm{aux},j} &= y + h\sum_{j=1}^{K} a_{j,l} f(t+\xi_j h, y_{\mathrm{aux},l}),\ j=1,...,K, \\ \Phi[f](t,y,h) &= \sum_{j=1}^{K} \beta_j f(t+\xi_j h, y_{\mathrm{aux},j}) \end{aligned} \tag{6.44} $$

qui résument la construction [13] de la fonctionnelle $\Phi[f]$ attachée au schéma numérique ainsi construit ; on calcule en effet, en dérivant ces formules par rapport à h et en combinant les résultats, l'expression de $\partial\Phi[f]/\partial h(t,y,0)$, soit

$$ \frac{\partial\Phi[f]}{\partial h}(t,h,0) = \sum_{j=1}^{K} \beta_j \Big(\xi_j \frac{\partial f}{\partial t}(t,y) + \frac{\partial f}{\partial y}(t,y)\, f(t,y) \big(\sum_{l=1}^{K} a_{j,l} \big) \Big) $$

(en effet, on utilise ici, une fois le calcul de dérivées par rapport à h opéré avec la règle de Leibniz, le fait que $(y_{\mathrm{aux},j})_{h=0} = y_{h=0} = y$). Il suffit d'écrire, pour que la méthode soit d'ordre 2, les égalités (6.27) pour $l = 0$ (on obtient la condition (6.36)) et $l = 1$ (on obtient alors les deux conditions additionnelles (6.43)).

Pour que le schéma construit à partir de cette méthode soit d'ordre au moins 3, il faut pousser plus loin les développements en exprimant que l'erreur de troncature $e_k(h;y_0)$ doit être dans ce cas en $0(h^3)$, ce qui conduit à calculer un développement de Taylor de cette erreur (en fonction de h). On ne détaillera pas ici ces calculs, nous contentant ici de renvoyer au chapitre 3 de [**MathAp**] (preuve du Théorème 3.26) où ils sont détaillés. Pour que la méthode soit d'ordre au moins 3 dans le cas particulier où

$$ \beta_j = \sum_{l=1}^{K} a_{j,l}\,,\ j=1,...,K \tag{6.45} $$

(autrement dit, la double quadrature fournit un calcul exact pour les fonctions constantes), il faut que soient réalisées, outre la condition de consistance (6.36), les trois conditions :

$$ \begin{aligned} {}^t\beta \cdot \mathrm{diag}\,[\xi_1,...,\xi_K] \cdot \mathtt{ones}\,[1,K] &= \frac{1}{2} \\ {}^t\beta \cdot \mathrm{diag}\,[\xi_1^2,...,\xi_K^2] \cdot \mathtt{ones}\,[1,K] &= \frac{1}{3} \\ {}^t\beta \cdot A \cdot \mathrm{diag}\,[\xi_1,...,\xi_K] \cdot \mathtt{ones}\,[1,K] &= \frac{1}{6}. \end{aligned} \tag{6.46} $$

Toujours sous les conditions (6.36) et (6.45), il faut et il suffit, pour que la méthode soit d'ordre 4 (c'est le cas de la méthode « classique » présentée dans les exemples

13. Elle se fait en deux temps : d'abord calculer les $y_{\mathrm{aux},j}$, $j=1,...,K$, de manière explicite ou implicite à partir de y ; ensuite reporter ces valeurs au membre de droite de la seconde ligne de (6.44) pour obtenir l'expression finale de $\Phi[f](t,y,h)$.

6.19 et 6.20 et la plus couramment utilisée), on doit avoir, outre les trois conditions (6.46), les quatre conditions additionnelles :

$$
\begin{aligned}
&{}^t\beta \cdot \mathrm{diag}\,[\xi_1^3, ..., \xi_K^3] \cdot \mathtt{ones}\,[1, K] = \frac{1}{4} \\
&{}^t\beta \cdot A \cdot \mathrm{diag}\,[\xi_1^2, ..., \xi_K^2] \cdot \mathtt{ones}\,[1, K] = \frac{1}{12} \\
&{}^t\beta \cdot A^2 \cdot \mathrm{diag}\,[\xi_1, ..., \xi_K] \cdot \mathtt{ones}\,[1, K] = \frac{1}{24} \\
&{}^t\beta \cdot \mathrm{diag}\,[\xi_1, ..., \xi_K] \cdot A \cdot \mathrm{diag}\,[\xi_1, ..., \xi_K] \cdot \mathtt{ones}\,[1, K] = \frac{1}{8}.
\end{aligned}
\tag{6.47}
$$

Et ainsi de suite ... On admettra ici ces résultats techniques, mais fort utiles, en retenant toutefois que la méthode « classique » est celle qui se révèle d'usage le plus fréquent.

REMARQUE 6.21 (les exemples de la section 6.3.1). Notons aussi que les méthodes d'Euler explicite, d'Euler implicite, et de Craig-Nicholson présentées à titre d'exemples préliminaires dans la section 6.3.1 sont des méthodes de Runge-Kutta à deux points ($\xi_1 = 0$, $\xi_2 = 1$) avec comme coefficients

$$a_{1,1} = a_{1,2} = 0, \quad a_{2,1} = 1 - \alpha \quad a_{2,2} = \alpha$$

et

$$\beta_1 = 1 - \alpha, \quad \beta_2 = \alpha$$

avec respectivement $\alpha = 0$ (Euler explicite, ordre 1), $\alpha = 1$ (Euler implicite, ordre 1), $\alpha = 1/2$ (Craig-Nicholson, implicite d'ordre 2).

FIN DU COURS

Bibliographie

[MathL2] J.M. Marco, P. Thieullen, J.P. Marco (ed.), Mathématiques L2, Pearson Education, 2007.

[MathAp] A. Yger & J.A. Weil (ed.), *Mathématiques Appliquées L3*, Pearson Education, Paris, 2009.

[Y1] A. Yger, *Calcul Symbolique et Scientifique*, polycopié de l'UE MHT304 :

`http://www.math.u-bordeaux1.fr/~yger/mht304.pdf`

[Zim] P. Zimmerman *Peut-on calculer sur ordinateur ?*, Leçon de Mathématiques d'aujourd'hui, Bordeaux, Octobre 2010 :

`http://www.loria.fr/~zimmerma/talks/lma.pdf`

Index