Calcul Numérique
Interpolation, Intégration, Équations Différentielles

François Boulier

9 septembre 2020

Introduction

Ce document constitue le support du cours de calcul numérique, à Polytech'Lille, pour les étudiants en cinquième année de la filière GIS (Génie Informatique et Statistique). Les notions abordées sont : l'interpolation polynomiale, l'intégration numérique et l'intégration numérique des équations différentielles ordinaires.

Un lien entre ces notions et la thématique GIS est donné par la modélisation des dynamiques de populations (de cellules, d'humains, de personnes atteintes par la grippe, par une opération de marketing viral, ...). Notons $y(x)$ la population d'une espèce au temps x. Le modèle le plus simple de dynamique de population est donné par une équation différentielle. Il s'agit du modèle de Malthus (1798), qui exprime l'idée qu'à tout instant x, la vitesse d'accroissement de la population est proportionnelle à la population existante :

$$\dot{y}(x) \quad = \quad k\, y(x)\,. \tag{1}$$

Le paramètre k est une constante de proportionalité. Voir [3, section 1.1, page 2]. Les solutions de ce modèle sont des exponentielles, qui tendent vers l'infini avec x. Ce modèle est donc inapplicable dans le cas où la population ne peut pas dépasser une valeur maximale, imposée par l'environnement, par exemple. Un modèle plus sophistiqué s'obtient en compliquant le membre droit de l'équation différentielle. Il s'agit du modèle suivant, appelé *logistique* par Verhulst (1838, 1845), qui l'a inventé :

$$\dot{y}(x) \quad = \quad k\, y(x) \left(1 - \frac{y(x)}{M}\right)\,. \tag{2}$$

Le paramètre M désigne la valeur maximale possible pour la population $y(x)$. Voir [3, section 1.1, page 3]. On peut facilement montrer que pour toute condition initiale $y(0) > 0$, la population $y(x)$ se rapproche de la valeur limite M.

Les modèles différentiels présentent tous l'inconvénient de ne lier l'évolution de la population au temps x qu'à la seule population existante et de ne pas tenir compte des populations passées. En particulier, les modèles exponentiels et logistiques ne tiennent absolument ps compte de la distribution des âges des populations [3, section 1.3, page 13] or, chez les humains, la vitesse de reproduction dépend fortement de l'âge et peut difficilement être représentée, de façon satisfaisante, par un simple paramètre k. Pour tenir compte de l'âge, il faut donc que l'accroissement de $y(x)$ dépende des populations $y(x-\tau)$ à tous les instants précédents $x-\tau$ et soit même la somme des contributions de ces populations. En passant à la limite, cette somme devient une intégrale. Voici

un exemple d'équation intégrale modélisant une dynamique de population humaine [2, section 2.1.1] :

$$y(x) \quad = \quad \alpha\, s(x) + k \int_0^x s(x-\tau)\, y(\tau)\, \mathrm{d}\tau\,. \tag{3}$$

Le paramètre α représente la population initiale. Le paramètre k agit sur la vitesse d'accroissement. La fonction $s(x)$ est une fonction de survie , c'est-à-dire une fonction à valeur dans $[0, 1]$, qui donne la fraction de la population née au temps 0, toujours vivante au temps x. Par conséquent, $s(x - \tau)$ donne la fraction de la population née au temps τ, toujours vivante au temps x (voir Figure **??**). Une telle fonction n'est bien sûr pas donnée par une formule mathématique mais peut avoir été établie par des mesures statistiques, sous la forme d'un tableau de valeurs.

Le cours commence par l'étude de méthodes d'interpolation (formule de Lagrange, différences divisées de Newton, splines). De telles méthodes peuvent être appliquées pour transformer le tableau de valeurs qui définit la fonction de survie en une fonction lisse, plus agréable à manipuler. Il continue par l'étude de méthodes d'intégration numérique (méthode des trapèzes, de Simpson). Combinées à des techniques d'algèbre linéaire, ces méthodes permettent de résoudre numériquement l'équation intégrale (3). La dernière partie du cours est consacrée aux méthodes numériques d'intégration des équations différentielles (schémas d'Euler, de Runge-Kutta), qui permettent de résoudre numériquement les modèles exponentiels (1), logistiques (2) et même ceux du type (3), pour peu qu'on fasse quelques hypothèses simplificatrices sur la fonction de survie [1, chapter II, section 17, page 351] et qu'on dérive l'équation intégrale pour obtenir une équation différentielle. On les illustre sur un problème d'astronomie [1] : le problème restreint à trois corps.

En 2019, ce support s'est enrichi d'un chapitre supplémentaire dédié aux splines lissantes.

Les travaux pratiques sont réalisés en FORTRAN, avec l'aide du logiciel de calcul formel MAPLE. Pour finir, je voudrais remercier Serge Petiton, qui a fondé ce cours, Mélanie Goguillot (GIS2A5 2015) pour avoir repéré une erreur dans le chapitre sur l'intégration numérique, Amélie Poilvert (GIS2A5 2016), Benoît Verhaeghe (GIS4 2016) pour avoir corrigé des formules dans le chapitre sur l'interpolation.

1. Aucun lien avec la dynamique de populations ? Pas si sûr : les modèles différentiels et intégraux mentionnés dans cette introduction ont été fortement étudiés par Vito Volterra (1860-1940). L'équation (3) est d'ailleurs une *équation de Volterra de la première espèce*. Mais c'est surtout après 1918 que Volterra s'est intéressé à la biologie. Plus jeune, il s'est intéressé à la physique mathématique. Il paraît qu'à l'âge de 13 ans, après la lecture du roman de Jules Verne *De la Terre à la Lune*, il a essayé de calculer la trajectoire d'un projectile dans le champ gravitationnel combiné de la Terre et de la Lune : c'est le problème restreint à trois corps. Quarante ans plus tard, en 1912, il démontra sa solution dans une série de cours à la Sorbonne [4, Biography of Vito Volterra, page 5].

,,

,,,,

Bibliographie

[1] Ernst Hairer, Syvert Paul Norsett, and Gerhard Wanner. *Solving ordinary differential equations I. Nonstiff problems*, volume 8 of *Springer Series in Computational Mathematics*. Springer–Verlag, New York, 2nd edition, 1993.

[2] Abdul J. Jerri. *Introduction to Integral Equations with Applications*, volume 93 of *Monographs and Textbooks in Pure and Applied Mathematics*. Marcel Dekker Inc., 1985.

[3] J. D. Murray. *Mathematical Biology I. An Introduction*, volume 17 of *Interdisciplinary Applied Mathematics*. Springer Verlag, third edition, 2002.

[4] Vito Volterra. *Theory of functionals and of integral and integro-differential equations*. Dover Publi. Inc., 1959. with a biography and a bibliography by Sir Edmund Whittaker.

Table des matières

Chapitre 1

Interpolation

polynomiale

1.1 Introduction

Par deux points distincts, il passe une unique droite. Cette observation se généralise. Le problème de l'interpolation s'exprime ainsi : étant donnés $n+1$ points

$$\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}, \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \ldots, \begin{pmatrix} x_n \\ y_n \end{pmatrix},$$

d'abscisses distinctes deux-à-deux, trouver l'unique polynôme $p(x)$ de degré au plus n dont le graphe passe par les $n+1$ points (dans le cas de 2 points, on a $n=1$ et le graphe d'un polynôme $p(x) = a_1\,x + a_0$ de degré 1 est bien une droite).

FIGURE 1.1 – Deux exemples d'interpolation avec 7 points équidistants. L'exemple de gauche montre que le polynôme d'interpolation peut permettre de reproduire assez fidèlement la courbe initiale dans l'intervalle d'interpolation mais très rarement en dehors de cet intervalle (un polynôme d'interpolation n'est pas un polynôme d'extrapolation). L'exemple de droite montre que, même dans l'intervalle d'interpolation, le graphe du polynôme peut avoir une forme très différente de celle de la courbe initiale.

les abscisses sont distincts deux-à-deux.

Donnée : $n+1$ points du le plan

$\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}, \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \ldots, \begin{pmatrix} x_n \\ y_n \end{pmatrix}$

On cherche :

les coefficients $a_0, a_1, \ldots, a_n$

d'un polynôme de degré (au plus n)

$p(x) = a_0 + a_1 x + \ldots + a_n x^n$

dont le graphe passe par chacun des points.

2 pts → 1 droite unique qui passe par les pts
= le graphe d'1 pol. de degré

0, 1, 2

→ aussi 1 parabole
= graphe pol. degré 2

de certains
ex. droite
pas 1 parabole

Impot : le polynôme d'interpolation est unique

ce pol s'appelle le polynôme d'interpolation.

D'un point de vue théorique, l'existence et l'unicité du polynôme d'interpolation s'établissent facilement. Il suffit d'attribuer des coefficients indéterminés $a_n, \ldots, a_1, a_0$ au polynôme d'interpolation $p(x) = a_n\,x^n + \cdots + a_1\,x + a_0$ et de poser que son graphe passe par les $n+1$ points :

n+1 inconnues

$$\begin{aligned} a_n\,x_0^n + \cdots + a_1\,x_0 + a_0 &= y_0\,, \\ a_n\,x_1^n + \cdots + a_1\,x_1 + a_0 &= y_1\,, \\ &\vdots \\ a_n\,x_n^n + \cdots + a_1\,x_n + a_0 &= y_n\,. \end{aligned}$$

plusieurs algos pour le calculer

ici qui calcule tous le ^ de Lagr

on pose que $p(x)$ passe par le graphe de $\begin{pmatrix} x_1 \\ y_n \end{pmatrix}$ $\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$

On aboutit à un système d'équations linéaires exactement déterminé :

$A \cdot x = b$

mat Vandermonde de

$$\underbrace{\begin{pmatrix} x_0^n & \ldots & x_0 & 1 \\ x_1^n & \ldots & x_1 & 1 \\ \vdots & & \vdots & \vdots \\ x_n^n & \ldots & x_n & 1 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} a_n \\ \vdots \\ a_1 \\ a_0 \end{pmatrix}}_{x} = \underbrace{\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}}_{b}.$$

Les coefficients de A sont des puissances des abscisses des $n+1$ points. Les coefficients de b sont les ordonnées. Comme les points sont supposés donnés, la matrice A et le vecteur b se calculent facilement. La matrice A a une forme particulière : c'est une *matrice de Vandermonde*. Cette matrice est inversible si et seulement si les x_i sont distincts deux-à-deux [4, Proposition 109, page 209]. Le système d'équations linéaires a donc une unique solution, qu'on peut calculer par factorisation $L\,U$ ou par factorisation $Q\,R$. Cette solution fournit les coefficients du polynôme $p(x)$ recherché.

Lcholesky

Fact. QR

On peut remarquer que le coefficient dominant, a_n, de $p(x)$ peut fort bien être nul. Dans ce cas, le polynôme $p(x)$ est de degré strictement inférieur [1] à n.

On pourrait aussi vouloir chercher un polynôme de degré strictement inférieur à n qui passe au plus près des $n+1$ points. On aboutit alors à des problèmes de moindres carrés, qui se mettent en équation de la même façon et se résolvent, eux-aussi, par factorisation $Q\,R$. Voir le support du cours de GIS3 [1, chapitres 4, 6 et 7].

1.2 À quoi l'interpolation sert-elle ?

On sait bien qu'il est facile d'évaluer un polynôme quand on connaît ses coefficients. En tant qu'outil théorique, l'interpolation résout le problème inverse et montre qu'un polynôme de degré n peut aussi bien se représenter par ses $n+1$ coefficients que par ses valeurs en $n+1$ points quelconques. Cette observation est la base des algorithmes les plus rapides connus pour la multiplication de polynômes.

L'interpolation apparaît aussi comme sous-problème dans de nombreux problèmes plus appliqués. Par exemple, pour intégrer une expression compliquée, il suffit de l'évaluer en $n+1$ points,

1. Par trois points, il passe en général une parabole mais, si les trois points sont alignés, la parabole dégénère en une droite.

d'approximer l'expression par un polynôme de degré n (interpolation) et d'intégrer ce polynôme. Cette observation est la base des schémas d'intégration numériques.

Enfin l'interpolation, en un sens plus large, où on ne cherche plus un polynôme mais d'autres types d'expressions mathématiques, est à la base de nombreuses techniques de compression des données (le format JPEG, par exemple). Pour plus de détails, on conseille la lecture de [2, chapitre II].

1.3 Erreur de l'interpolation

Supposons que les $n+1$ points appartiennent au graphe d'une fonction $y = f(x)$. On s'intéresse à l'erreur de l'interpolation, c'est-à-dire l'écart entre le graphe de la fonction et celui du polynôme d'interpolation $p(x)$. Voir Figure 1.1 pour des exemples.

Le théorème 1 suivant permet de majorer l'écart entre le graphe de la fonction et celui du polynôme d'interpolation, même en dehors de l'intervalle d'interpolation $[\min x_i, \max x_i]$! Il fait intervenir la dérivée $(n+1)$-ième de $f(x)$, notée $f^{(n+1)}(x)$. Il implique que si $f(x)$ est un polynôme de degré n alors, $p(x) = f(x)$. Un exemple, montrant comment utiliser le théorème, est donné Figure 1.2.

Théorème 1 *Supposons $f(x)$ définie sur un intervalle $[a, b]$ (pas nécessairement inclus dans l'intervalle d'interpolation) et $n+1$ fois dérivable sur cet intervalle. Alors, pour tout $x \in [a, b]$, il existe $\xi \in [\min(x_i, x), \max(x_i, x)]$ tel que*

$$f(x) - p(x) = (x - x_0)(x - x_1) \cdots (x - x_n) \frac{f^{(n+1)}(\xi)}{(n+1)!}.$$

Preuve. Voir [2, Théorème 2.2, page 27]. □

1.4 Calcul du polynôme d'interpolation

Les $n + 1$ points étant fixés, le polynôme d'interpolation $p(x)$ est défini de façon unique. Il existe plusieurs méthodes pour le déterminer. Parmi celles-ci :

— la formule de Lagrange [2], conceptuellement très simple ;
— l'algorithme de Neville [3] ;
— la technique des différences divisées de Newton [4].

Aussi bien l'algorithme de Neville que les différences divisées de Newton permettent d'évaluer le polynôme d'interpolation $p(x)$ sans calculer ses coefficients $a_n, \ldots, a_1, a_0$, ce qui est avantageux numériquement. L'algorithme de Neville est plus rapide si on cherche les valeurs de $p(x)$ pour un nombre de valeurs de x inférieur à n. Dans le cas contraire, les différences divisées de Newton sont plus efficaces parce qu'on peut les précalculer avant de procéder aux évaluations.

2. Joseph-Louis Lagrange (1736-1813).
3. Eric Harold Neville (1889-1961).
4. Isaac Newton (1643-1727).

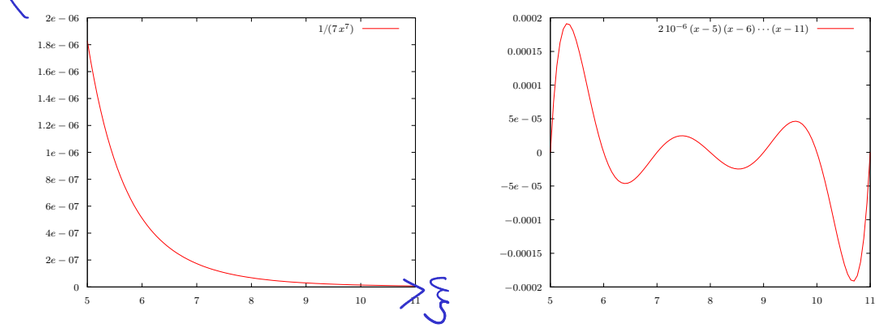

FIGURE 1.2 – Illustration du thèorème 1 sur l'exemple de la Figure 1.1, à gauche. On a $f(x) = \ln(x)$ et $n = 6$. On a pris $[a, b]$ égal à l'intervalle d'interpolation $[5, 11]$. En traçant le graphe de $f^{(7)}(\xi)/7! = 1/(7\,\xi^7)$ pour $\xi \in [a, b]$, on trouve que cette expression est majorée par $2\,10^{-6}$ (à gauche). D'après le théorème, sur l'intervalle $[a, b]$, on a donc $|f(x) - p(x)| \leqslant |2\,10^{-6}\,(x-5)\,(x-6)\cdots(x-11)|$. En traçant le graphe de cette expression, on trouve que l'erreur ne dépasse par $2\,10^{-4}$ (à droite).

1.4.1 La formule de Lagrange

On cherche à exprimer le polynôme d'interpolation sous la forme d'une somme :

$$p(x) \quad = \quad \sum_{i=0}^{n} L_i(x)\,.$$

On cherche donc un polynôme $L_i(x)$ pour chacun des points (x_i, y_i). Il suffit donc que ces $n+1$ polynômes $L_i(x)$ vérifient :

$$L_i(x_i) = y_i\,, \quad L_i(x_j) = 0 \quad (\text{pour } j \neq i)\,.$$

Fixons un indice $0 \leqslant i \leqslant n$. La condition $L_i(x_j) = 0$ (pour $j \neq i$) impose que $L_i(x)$ soit de la forme suivante, où c_i est un réel qui reste à déterminer :

$$L_i(x) \quad = \quad c_i\,(x - x_0)\cdots(x - x_{i-1})\,(x - x_{i+1})\cdots(x - x_n)\,.$$

La condition $L_i(x_i) = y_i$ impose alors[5] la valeur de c_i :

$$c_i \quad = \quad \frac{y_i}{(x_i - x_0)\cdots(x_i - x_{i-1})\,(x_i - x_{i+1})\cdots(x_i - x_n)}.$$

5. Le fait que les abscisses soient distinctes deux-à-deux évite une division par zéro.

$L_1(x) = \frac{q}{?} \cdot (x-x_0)(x-x_2)$ comme $L_1(x_0) = L_1(x_2) = 0$

$\frac{y_1}{(x_1-x_0)(x_1-x_2)}$

Le coef qu'il faut multiplier pour que $L_1(x_1) = y_1$

$y_1 \dfrac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}$ ← pol de degré 2

$\{$ non $\neq 0$ car rapport vaut 1 qd $x = x_1$

1.4.2 L'algorithme de Neville

Commençons par la recette et par un exemple. L'algorithme se résume à la récurrence suivante. Le polynôme d'interpolation recherché se trouve en $p_{0,n}$.

$$\begin{aligned} p_{i,0}(x) &= y_i && (0 \leqslant i \leqslant n), \\ p_{i,j}(x) &= \frac{(x_i - x)\, p_{i+1,j-1}(x) + (x - x_{i+j})\, p_{i,j-1}(x)}{x_i - x_{i+j}} && (1 \leqslant j \leqslant n,\quad 0 \leqslant i \leqslant n-j). \end{aligned}$$

inconnue

Voici un exemple ($n = 3$) :

	0	1	2	3
x_i	1	2	3	5
y_i	1	4	2	5

idée : on veut tracer le graphe de $P(x)$
on n'explicite jamais les coeffs de P.
on fait varier x sur un intervalle → x
Pour chq x (= flottant) on remplit le tableau. On a en haut à droite la valeur de $P(x)$ en x.

Les polynômes $p_{i,j}$ peuvent être organisés suivant un tableau. Le calcul se fait colonne par colonne. Chaque case $p_{i,j}$ se calcule à partir de la case $p_{i,j-1}$ située à sa gauche et de la case $p_{i+1,j-1}$, située en bas, à gauche.

i \ j	0	1	2	3
0	$1 \to 3x-2$	$-2x+8 \to \frac{7}{6}x^2 - \frac{47}{6}x + 15$	$-\frac{5}{2}x^2 + \frac{21}{2}x - 7$	$\frac{11}{12}x^3 - 8x^2 + \frac{247}{12}x - \frac{25}{2}$
1	4	$\frac{3}{2}x - \frac{5}{2}$		
2	2			
3	5			

$p_{i,j-1}$; $p_{i,j}$; $p_{i+1,j-1}$

$f(x) = 3x - 2$
$f(1) = 1$ (x_1)
$f(2) = 6$ (x_2)

idée : chq $p_{i,j}$ est lui-même le pol d'interpolation le pol dont le graphe passe par $\binom{x_i}{y_i}, \dots, \binom{x_{i+j}}{y_{i+j}}$

présenté sa forme symbolique

Numériquement, l'indéterminée x est un nombre à virgule flottante. Pour chaque valeur de x, on remplit la matrice et on trouve la valeur du polynôme d'interpolation pour cette valeur de x en $p_{0,n}$.

3 boucles — sur x — extérieure
sur i — intérieure
sur j — intermédiaire

1.4.3 Explication

L'argumentation ci-dessous justifie non seulement l'algorithme précédent mais permet aussi d'introduire les différences divisées de Newton[6]. Elle est fortement inspirée de [3].

Oublions temporairement la relation de récurrence précédente et définissons $p_{i,j}$ comme le polynôme d'interpolation dont le graphe passe par les $j+1$ points d'indices $i, \dots, i+j$. Supposons pour simplifier que $p_{i,j}$ soit de degré j exactement. Notons c son coefficient dominant. Alors on a :

$$p_{i,j} - p_{i,j-1} = c\,(x - x_i) \cdots (x - x_{i+j-1})\,. \tag{1.1}$$

Preuve. Il suffit de montrer que la différence $p_{i,j} - p_{i,j-1}$ est un polynôme de degré j, de coefficient dominant c, ayant j racines $x_i, \dots, x_{i+j-1}$. Les deux polynômes sont de degrés respectifs j et $j-1$. La différence des deux est donc de degré j et a un coefficient dominant égal à c. Comme $p_{i,j}$ et $p_{i,j-1}$ sont deux polynômes d'interpolation ayant les j points d'indices i à $i+j-1$ en commun, leur différence s'annule sur les abscisses de ces j points. □

6. Neville a vécu 200 ans après Newton.

Par un raisonnement similaire,

$$p_{i,j} - p_{i+1,j-1} \quad = \quad c\,(x - x_{i+1}) \cdots (x - x_{i+j})\,. \tag{1.2}$$

Par conséquent,

$$\frac{p_{i,j} - p_{i,j-1}}{p_{i,j} - p_{i+1,j-1}} = \frac{x - x_i}{x - x_{i+j}}\cdot$$

Il suffit de tirer $p_{i,j}$ en fonction des autres expressions pour retrouver la récurrence de Neville.

Le même type de raisonnement, permet de calculer les coefficients dominants (notons-les $c_{i,j}$) des polynômes $p_{i,j}$ plutôt que les polynômes eux-mêmes. On a :

$$p_{i+1,j-1} - p_{i,j-1} \quad = \quad (c_{i+1,j-1} - c_{i,j-1})\,(x - x_{i+1}) \cdots (x - x_{i+j-1})\,. \tag{1.3}$$

Preuve. Les deux polynômes ont même degré, $j-1$. Le coefficient dominant de leur différence est donc égal à la différence de leurs coefficients dominants. Ce sont deux polynômes d'interpolation ayant les $j-1$ points d'indices $i+1$ à $i+j-1$ en commun. Leur différence s'annule donc sur les abscisses de ces $j-1$ points. □

Considérons l'égalité évidente

$$\underbrace{p_{i,j} - p_{i,j-1}}_{(1.1)} \quad = \quad \underbrace{p_{i,j} - p_{i+1,j-1}}_{(1.2)} + \underbrace{p_{i+1,j-1} - p_{i,j-1}}_{(1.3)}\,.$$

En remplaçant les expressions soulignées par les formules correspondantes, on trouve :

$$\begin{aligned} c_{i,j}\,(x - x_i) \cdots (x - x_{i+j-1}) \quad &= \quad c_{i,j}\,(x - x_{i+1}) \cdots (x - x_{i+j}) \\ &\quad + (c_{i+1,j-1} - c_{i,j-1})\,(x - x_{i+1}) \cdots (x - x_{i+j-1}) \end{aligned}$$

Divisons les deux membres de l'égalité par $(x - x_{i+1}) \cdots (x - x_{i+j-1})$. On obtient :

$$c_{i,j}\,(x - x_i) \quad = \quad c_{i,j}\,(x - x_{i+j}) + (c_{i+1,j-1} - c_{i,j-1})$$

et donc

$$c_{i,j} \quad = \quad \frac{c_{i+1,j-1} - c_{i,j-1}}{x_{i+j} - x_i}\cdot \tag{1.4}$$

Ces coefficients $c_{i,j}$ sont les différences divisées de Newton.

1.4.4 Les différences divisées de Newton

La méthode de Newton consiste à calculer le tableau des différences divisées $c_{i,j}$ en utilisant la récurrence (1.4) :

$$\begin{aligned} c_{i,0} \quad &= \quad y_i \qquad\qquad\qquad\qquad (0 \leqslant i \leqslant n), \\ c_{i,j} \quad &= \quad \frac{c_{i+1,j-1} - c_{i,j-1}}{x_{i+j} - x_i} \qquad (1 \leqslant j \leqslant n\,, \quad 0 \leqslant i \leqslant n - j). \end{aligned}$$

Les coefficients peuvent être organisés suivant un tableau, comme pour l'algorithme de Neville. La grande différence par rapport à celui-ci, c'est que le tableau ne dépend pas de x et peut se calculer une fois pour toutes. Sur l'exemple précédent, on trouve :

i \ j	0	1	2	3
0	1	3	$-\frac{5}{2}$	$\frac{11}{12}$
1	4	-2	$\frac{7}{6}$	
2	2	$\frac{3}{2}$		
3	5			

Algo de Newton
On ne mémorise que les coeffs dominants des polynômes de Neville
On peut les calculer une fois pour toutes.
Connaissant la diagonale on peut évaluer le polynôme d'interpolation

Il ne reste plus qu'à reconstruire le polynôme d'interpolation $p_{0,n}$ à partir des coefficients. On se sert pour cela de la relation (1.2), qui exprime le polynôme $p_{i,j}$ en fonction du polynôme $p_{i+1,j-1}$ situé sur la case en bas à gauche. En remontant la diagonale, on trouve :

$$
\begin{aligned}
p_{n,0} = p_{3,0} &= c_{3,0} \\
&= 5\,, \\
p_{2,1} &= c_{3,0} + c_{2,1}\,(x - x_3) \\
&= 5 + \frac{3}{2}\,(x-5)\,, \\
p_{1,2} &= c_{3,0} + c_{2,1}\,(x - x_3) + c_{1,2}\,(x - x_3)\,(x - x_2) \\
&= 5 + \frac{3}{2}\,(x-5) + \frac{7}{6}\,(x-5)\,(x-3)\,, \\
p_{0,n} = p_{0,3} &= c_{3,0} + c_{2,1}\,(x - x_3) + c_{1,2}\,(x - x_3)\,(x - x_2) + c_{0,3}\,(x - x_3)\,(x - x_2)\,(x - x_1) \\
&= 5 + \frac{3}{2}\,(x-5) + \frac{7}{6}\,(x-5)\,(x-3) + \frac{11}{12}\,(x-5)\,(x-3)\,(x-2)\,.
\end{aligned}
$$

L on peut le faire de façon efficace avec une variante du schéma de Horner-Ruffini

Algo 1) 2 boucles imbriquées pour remplir le tableau
2) 1 boucle qui fait varier $x \in$ xplst
L imbriqué : 1 boucle pour le schéma de Horner-Ruffini

On a bien sûr tout intérêt à utiliser un schéma de Horner[7] (le pseudo-code correspondant se trouve en Figure 1.3) :

$$p_{0,3} = ((c_{0,3}\,(x - x_1) + c_{1,2})\,(x - x_2) + c_{2,1})\,(x - x_3) + c_{3,0}\,.$$

1.4.5 Conclusion

La technique des différences divisées de Newton est plus efficace que l'algorithme de Neville lorsqu'on a besoin d'évaluer le polynôme d'interpolation pour un grand nombre de valeurs de x. Dans les deux cas, on obtient des formules un peu plus simples (mais des explications un peu plus compliquées) en transformant les récurrences pour calculer des matrices triangulaires inférieures plutôt que supérieures.

7. Il s'agit d'un schéma de Horner un peu particulier. Le schéma de Horner classique est une formule permettant d'évaluer un polynôme de degré n au moyen de n multiplications, alors qu'une évaluation naive en demande $n\,(n+1)/2$. Le schéma de Horner est résumé par la formule suivante (cas $n = 4$) :

$$a_4\,x^4 + a_3\,x^3 + a_2\,x^2 + a_1\,x + a_0 = (((a_4\,x + a_3)\,x + a_2)\,x + a_1)\,x + a_0\,.$$

Il est nommé d'après William George Horner (1819-1845). On l'appelle aussi parfois : schéma de Ruffini-Horner.

$p = c_{0,n}$
for $i = n-1, n-2, \ldots, 0$ do
  $p = p\,(x - x_{n-i}) + c_{n-i,i}$
end do

FIGURE 1.3 – Schéma de Horner évaluant le polynôme d'interpolation $p_{0,n}$ à partir de la matrice $(c_{i,j})$ des différences divisées.

1.5 Les splines

Les *splines* interpolantes sont des variantes de polynômes d'interpolation. L'idée consiste à associer, à chaque intervalle $[x_i, x_{i+1}]$, un polynôme $s_i(x)$ de degré 3 (la spline est alors dite *cubique*) dont le graphe passe par les deux points d'indices i et $i+1$, tout en imposant des conditions sur les dérivées $s_i'(x)$ et $s_i''(x)$ pour que le raccord entre les graphes de $s_i(x)$ et de $s_{i+1}(x)$ soit suffisamment lisse (voir ci-dessous). Il existe des splines non interpolantes, qui permettent de lisser des suites de points. Elles sont décrites en détail au chapitre 3.

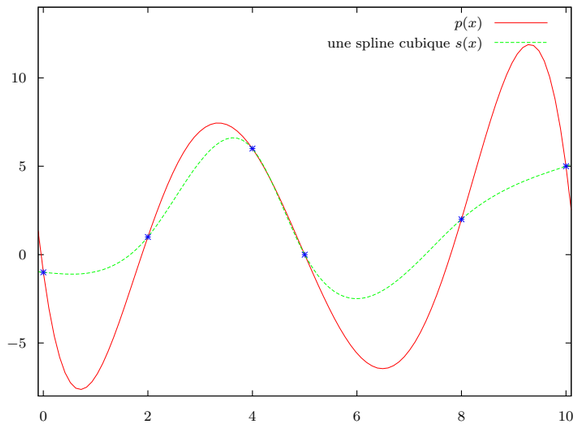

FIGURE 1.4 – Le polynôme $p(x)$ est un polynôme d'interpolation. Son graphe est C^∞. La fonction $s(x)$ est une spline cubique interpolante, c'est-à-dire une fonction polynomiale de degré 3 par morceaux, dont le graphe est globalement C^2, passant par les mêmes points.

Le qualificatif « lisse » peut avoir plusieurs sens, qu'on peut préciser mathématiquement : une courbe est dite C^0 si elle est continue ; elle est dite C^1 si elle est C^0 et si sa dérivée première est continue ; elle est dite C^2 si elle est C^1 et si sa dérivée seconde est continue. Voir Figure 1.5.

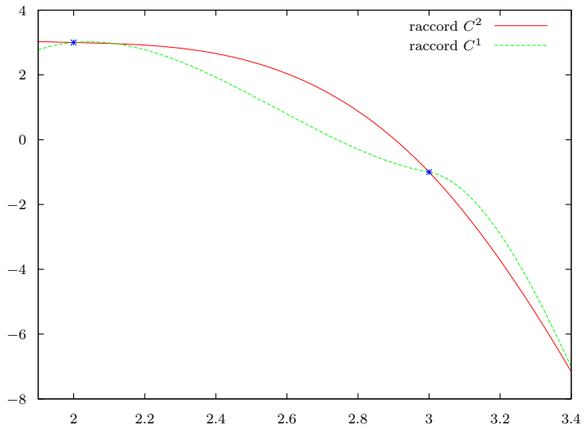

FIGURE 1.5 – La différence entre un raccord C^1 et un raccord C^2 peut s'apprécier visuellement. Le raccord a lieu au point $(3, -1)$.

Idée. Si on associe à chaque point une pente arbitraire y'_i $(0 \leqslant i \leqslant n)$, il est facile d'obtenir des polynômes $s_i(x)$ de degré 3, qui passent par les points d'indices i et $i+1$ en suivant ces pentes. On obtient alors une fonction définie par morceaux dont le graphe est C^1.

Une spline cubique est aussi une fonction définie par morceaux mais son graphe est C^2. Les dérivées y'_i et y''_i ne peuvent alors plus être choisies arbitrairement, sauf la première y'_0 et la dernière y'_n : elles sont la solution d'un système d'équations linéaires.

Le graphe d'une spline cubique semble plus « doux » que celui d'un polynôme d'interpolation (c'est la raison d'être des splines) mais, mathématiquement, le graphe d'un polynôme d'interpolation est plus lisse que celui d'une spline, puisqu'il est C^∞.

1.5.1 Construction d'une courbe C^1

Supposons données les pentes y'_0 et y'_1, c'est-à-dire les valeurs que doivent prendre $s'_0(x_0)$ et $s'_0(x_1)$. Pour obtenir le polynôme $s_0(x)$, il suffit de lui attribuer des coefficients indéterminés : $s_0(x) = a_3\,x^3 + a_2\,x^2 + a_1\,x + a_0$ et d'écrire les conditions

$$s_0(x_0) = y_0\,, \quad s'_0(x_0) = y'_0\,, \quad s_0(x_1) = y_1\,, \quad s'_0(x_1) = y'_1\,.$$

On obtient alors un système d'équations linéaires exactement déterminé qui se résout par factorisation $L\,U$ ou factorisation $Q\,R$:

$$\underbrace{\begin{pmatrix} x_0^3 & x_0^2 & x_0 & 1 \\ 3\,x_0^2 & 2\,x_0 & 1 & 0 \\ x_1^3 & x_1^2 & x_1 & 1 \\ 3\,x_1^2 & 2\,x_1 & 1 & 0 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{pmatrix}}_{x} = \underbrace{\begin{pmatrix} y_0 \\ y_0' \\ y_1 \\ y_1' \end{pmatrix}}_{b}.$$

On détermine le polynôme $s_1(x)$ de la même façon. Pour que le raccord entre les graphes de $s_0(x)$ et $s_1(x)$ soit lisse, il faut bien sûr que $s_0'(x_1) = s_1'(x_1)$.

Exemple. Les données suivantes donnent une courbe C^1 visuellement proche de la courbe de la Figure 1.4. Les y_i' ont été choisis arbitrairement.

x_i	0	2	4	5	8	10
y_i	-1	1	6	0	2	5
y_i'	-1	3	-2	-2	2	1

La fonction $s(x)$ correspondante (qui n'est pas une spline) est :

$$s(x) = \begin{cases} x^2 - x - 1 & (x \leqslant 2) \\ -x^3 + \frac{31}{4}\,x^2 - 16\,x + 10 & (x \leqslant 4) \\ 8\,x^3 - 108\,x^2 + 478\,x - 690 & (x \leqslant 5) \\ -\frac{4}{27}\,x^3 + \frac{32}{9}\,x^2 - \frac{238}{9}\,x + \frac{1670}{27} & (x \leqslant 8) \\ -\frac{1}{4}\,x^2 + 6\,x - 30 & \text{sinon} \end{cases}$$

Exemple. Si on prend les données suivantes (les coordonnées des points sont les mêmes, seules les pentes ont changé), on obtient une courbe C^2 : c'est la spline cubique de la Figure 1.4. Les y_i' ont été obtenus en résolvant un système d'équations linéaires.

x_i	0	2	4	5	8	10
y_i	-1	1	6	0	2	5
y_i'	$-\frac{656}{2283}$	$\frac{8161}{2283}$	$-\frac{16033}{4566}$	$-\frac{50255}{9132}$	$\frac{11687}{4566}$	$\frac{2215}{2283}$

1.5.2 Condition pour que la courbe soit C^2

Étant donnés $n+1$ points, on cherche n polynômes s_i de degré 3 (quatre coefficients chacun) : un pour chaque intervalle. On doit donc trouver $4\,n$ coefficients indéterminés. Il s'agit donc de trouver $4\,n$ équations linéaires pour que le système à résoudre soit exactement déterminé. Le graphe de chaque polynôme s_i doit passer par les points d'indices $i-1$ et i. Cela fait $2\,n$ équations.

Les dérivées premières et secondes de deux polynômes s_i et s_{i+1} doivent coïncider au i-ème point. Cela fait $2\,(n-1)$ nouvelles équations. Il reste deux équations à poser. On les obtient en imposant des conditions aux deux extrémités.

On peut par exemple poser que $s_1''(x_0) = s_n''(x_n) = 0$. La spline est dite alors *naturelle*.

Lorsqu'on cherche à modéliser un phénomène périodique, on peut aussi rajouter une cubique s_{n+1} pour relier le point n au point 0. On obtient une spline dite alors *périodique*.

Bibliographie

[1] François Boulier. Calcul Numérique. Algèbre Linéaire. Technical report, Université Lille 1, 59655, Villeneuve d'Ascq, France, 2013. Support de cours de GIS3, accessible sur `http://cristal.univ-lille.fr/~boulier`.

[2] Ernst Hairer and Gerhard Wanner. Introduction à l'Analyse Numérique. Accessible sur `http://www.unige.ch/~hairer`, juin 2005.

[3] Wikipedia. Algorithme de Neville. `fr.wikipedia.org/wiki/Algorithme_de_Neville`.

[4] Richard Zippel. *Effective Polynomial Computation*. Springer Science + Business Media LLC, 1993.

Chapitre 2

Intégration Numérique

2.1 Introduction

Étant donnés un intervalle $[a, b]$ et une fonction $f(x)$, continue sur $[a, b]$, on cherche à approximer l'intégrale suivante. Voir Figure 2.1. Les formules approchées utilisées sont appelées *formules de quadrature*.

$$\int_a^b f(x)\,\mathrm{d}x\,. \tag{2.1}$$

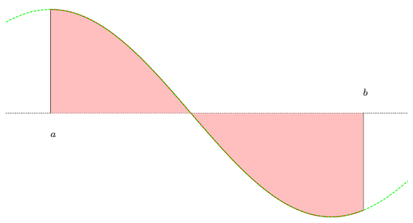

FIGURE 2.1 – L'intégrale (2.1) est un nombre signé : c'est la différence entre l'aire hachurée située au-dessus de l'axe horizontal et celle située au-dessous.

Remarque 1. Si on connaît une *primitive* $F(x)$ de $f(x)$, c'est-à-dire une fonction $F(x)$ dont la dérivée, $F'(x)$, est égale à $f(x)$, alors le problème posé se résout facilement, puisque

$$\int_a^b f(x)\,\mathrm{d}x = [F(x)]_a^b = F(b) - F(a)\,.$$

Remarque 2. Si $f(x)$ est un polynôme, alors une primitive $F(x)$ se calcule facilement et le problème posé se résout de même (voir remarque 1).

Cas général. Mais si $f(x)$ est une fonction compliquée, dont on ne connaît aucune primitive, ou même une fonction simple (comme $\sin(x)$) mais donnée sous la forme d'une fonction FORTRAN qui l'évalue (situation courante en calcul numérique) ? Il suffit d'appliquer l'idée suivante !

Idée $(*)$

1. fixer un entier n, évaluer $f(x)$ en $n+1$ points avec $x_0 = a$, $x_n = b$ et $y_i = f(x_i)$:
$$\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}, \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \ldots, \begin{pmatrix} x_n \\ y_n \end{pmatrix} ; \tag{2.2}$$
2. calculer le polynôme d'interpolation $p(x)$ dont le graphe passe par les $n+1$ points (le graphe du polynôme d'interpolation approxime le graphe de $f(x)$) ;
3. intégrer le polynôme d'interpolation (remarques 1 et 2).

Remarque 3. Pour appliquer l'idée $(*)$, il n'est pas nécessaire de connaître $f(x)$: il suffit de connaître les points (2.2). Cette remarque est particulièrement intéressante lorsque les points proviennent d'expériences, sont produits par des appareils de mesure ...

2.2 Les méthodes composites et leur ordre

On suppose une suite de points (2.2) connue pour un certain n fixé. On suppose de plus les abscisses x_i équidistantes et on note $h = x_{i+1} - x_i$ la longueur d'un pas. Comme on va le voir, l'idée $(*)$ explique comment les formules de Newton-Cotes [1] sont construites [2].

2.2.1 La formule des trapèzes composites

Elle consiste à approximer l'intégrale (2.1) en sommant les aires [3] des n trapèzes définis par les deux points d'indice i et $i+1$, pour $0 \leqslant i < n$. Cette méthode est en fait une application de l'idée $(*)$. Considérons les deux premiers points (x_0, y_0) et $(x_1, y_1) = (x_0 + h, y_1)$. L'aire du trapèze défini par ces deux points n'est rien d'autre que

$$\int_{x_0}^{x_0+h} p_{01}(x)\,\mathrm{d}x$$

où $p_{01}(x)$ désigne la droite passant par ces deux points. Le calcul suivant, effectué en MAPLE, retrouve la formule de quadrature bien connue :

1. Roger Cotes (1682-1716) est un contemporain d'Isaac Newton.
2. Pour rassurer les étudiants stressés : il n'est pas nécessaire de se rappeler comment les formules sont construites pour savoir les utiliser :-).
3. Ce sont des aires généralisées, où on compte positivement l'aire au-dessus de l'axe horizontal et négativement l'aire au-dessous.

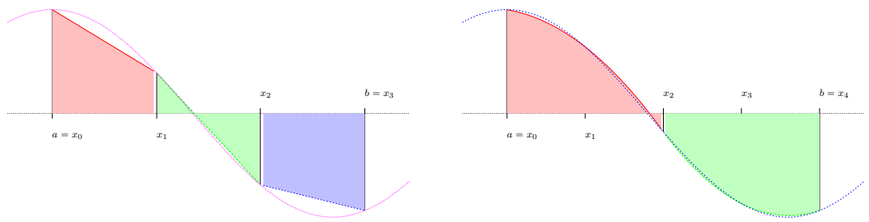

FIGURE 2.2 – Approximation de l'intégrale (2.1) par la méthode des trapèzes avec $n = 3$ intervalles (à gauche) et par la méthode de Simpson avec $n = 4$ intervalles (à droite).

```
> with (CurveFitting):
> p01 := PolynomialInterpolation ([[x0,y0],[x0+h,y1]], x);
                          (y1 - y0) x   y0 h - y1 x0 + y0 x0
                 p01 := ------------ + --------------------
                              h                  h
```

```
> simplify (int (p01, x=x0..x0+h));
                                    h (y0 + y1)
                                    -----------
                                         2
```

La même formule (aux indices près) s'applique à chaque trapèze. En sommant les expressions obtenues, on obtient la formule des trapèzes composites [4] :

$$\begin{aligned}\int_a^b f(x)\,\mathrm{d}x &\simeq \frac{h}{2}\left((y_0+y_1)+(y_1+y_2)+\cdots+(y_{n-1}+y_n)\right)\\ &= h\left(\frac{y_0}{2}+y_1+y_2+\cdots+y_{n-1}+\frac{y_n}{2}\right).\end{aligned}$$

2.2.2 La formule de Simpson composite

Encore une application de l'idée $(*)$. On considère les trois premiers points (x_0, y_0), $(x_1, y_1) = (x_0 + h, y_1)$ et $(x_2, y_2) = (x_0 + 2\,h, y_2)$. On calcule le polynôme d'interpolation p_{02} de degré au plus 2 dont le graphe (une parabole) passe par ces trois points et on approxime l'intégrale recherchée par l'intégrale, sur deux intervalles, du polynôme d'interpolation (voir Figure 2.2) :

$$\int_{x_0}^{x_0+2\,h} p_{02}(x)\,\mathrm{d}x$$

4. Une formule *composite* est une formule de quadrature, itérée sur une suite d'intervalles.

Le calcul de la formule de quadrature est pénible à la main mais s'effectue très facilement en MAPLE :

```
> p02 := PolynomialInterpolation ([[x0,y0],[x0+h,y1],[x0+2*h,y2]], x):
> simplify (int (p02, x=x0..x0+2*h));
                              h (y0 + y2 + 4 y1)
                              ------------------
                                      3
```

La même formule s'applique aux trois points (x_2, y_2), $(x_3, y_3) = (x_2 + h, y_2)$ et $(x_4, y_4) = (x_2 + 2\,h, y_4)$. Si on suppose n pair, elle s'applique plus généralement à chaque suite de deux intervalles et, en sommant les expressions obtenues, on obtient la formule de Simpson [5] composite :

$$\begin{aligned}\int_a^b f(x)\,\mathrm{d}x &\simeq \frac{h}{3}\,((y_0 + 4\,y_1 + y_2) + (y_2 + 4\,y_3 + y_4) + \cdots + (y_{n-2} + 4\,y_{n-1} + y_n))\\ &= \frac{h}{3}\,(y_0 + 4\,y_1 + 2\,y_2 + 4\,y_3 + \cdots + 2\,y_{n-2} + 4\,y_{n-1} + y_n)\ .\end{aligned}$$

2.2.3 Formules de Newton-Cotes

Cette section doit beaucoup à [2, chapitre I]. Les formules de quadrature des trapèzes et de Simpson sont des cas particuliers des formules de Newton-Cotes, qui sont obtenues en généralisant l'idée $(*)$ à un nombre d'intervalles quelconque. Pour les présenter sous leur forme générale, il est souhaitable de les réécrire légèrement, en mettant en facteur la longueur de l'intervalle d'intégration (h dans le cas des trapèzes, $2\,h$ dans le cas de Simpson) :

$$\begin{aligned}\int_{x_0}^{x_0+h} f(x)\,\mathrm{d}x &\simeq h\left(\frac{1}{2}\,y_0 + \frac{1}{2}\,y_1\right) && \text{(trapèzes)}\\ \int_{x_0}^{x_0+2\,h} f(x)\,\mathrm{d}x &\simeq 2\,h\left(\frac{1}{6}\,y_0 + \frac{4}{6}\,y_1 + \frac{1}{6}\,y_2\right) && \text{(Simpson)}\end{aligned}$$

On voit qu'avec cette convention, les formules de quadrature pour lesquelles les abscisses x_i sont équidistantes, sont complètement déterminées par la liste des coefficients numériques des ordonnées y_i. Un tableau récapitulatif est donné Figure 2.3.

Définition 1 *une formule de quadrature à s étages est une formule :*

$$\int_{x_0}^{x_0+(s-1)\,h} f(x)\,\mathrm{d}x \simeq (s-1)\,h\sum_{i=0}^{s-1} b_i\,y_i\,.$$

Les coefficients numériques b_i sont les poids *de la formule.*

Définition 2 *Une formule de quadrature est dite d'ordre p si elle est exacte pour tous les polynômes de degré strictement inférieur à p.*

5. Thomas Simpson (1710-1761).

s	ordre	poids b_i							nom
2	2	$\frac{1}{2}$	$\frac{1}{2}$						trapèzes
3	4	$\frac{1}{6}$	$\frac{4}{6}$	$\frac{1}{6}$					Simpson
4	4	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$				Newton
5	6	$\frac{7}{90}$	$\frac{32}{90}$	$\frac{12}{90}$	$\frac{32}{90}$	$\frac{7}{90}$			Boole
6	6	$\frac{19}{288}$	$\frac{75}{288}$	$\frac{50}{288}$	$\frac{50}{288}$	$\frac{75}{288}$	$\frac{19}{288}$		
7	8	$\frac{41}{840}$	$\frac{216}{840}$	$\frac{27}{840}$	$\frac{272}{840}$	$\frac{27}{840}$	$\frac{216}{840}$	$\frac{41}{840}$	Weddle

FIGURE 2.3 – Formules de Newton-Cotes pour $s \leqslant 7$.

Remarque. Le fait qu'une formule à s étages soit exacte pour les polynômes de degré $s-1$ est une conséquence immédiate de l'unicité des polynômes d'interpolation. Toute formule à s étages est donc d'ordre au moins s. On remarque toutefois que les formules de Newton-Cotes sont toujours d'ordre pair. C'est une conséquence du fait que les abscisses sont équidistantes et que les formules de Newton-Cotes sont symétriques (c'est-à-dire : $b_i = b_{s-i-1}$ pour $0 \leqslant i < s$). Pour une démonstration, générale voir [2, Théorème 1.4]. Les instructions MAPLE ci-dessous démontrent que la formule de Simpson est bien d'ordre 4. On affecte à f un polynôme général de degré 3.

```
> f := add (c[i]*x^i, i = 0 .. 3);
                                        2         3
                  f := c[0] + c[1] x + c[2] x  + c[3] x
```

On l'intègre entre a et b.

```
> I_exacte := int (f, x = a .. b);
                                      2    2              3    3
I_exacte := c[0] (b - a) + 1/2 c[1] (b  - a ) + 1/3 c[2] (b  - a )

                   4    4
     + 1/4 c[3] (b  - a )
```

On pose que n est pair et on calcule une approximation de l'intégrale par la formule de Simpson composite.

```
> n := 2*m:
> h := (b-a)/n:
> I_Simpson := h/3 * (eval (f, x=a) + eval (f, x=b) +
                      4*sum (eval (f, x=a+(2*i+1)*h), i=0..n/2-1) +
                      2*sum (eval (f, x=a+2*i*h), i=1..n/2-1));
```

```
                                                          2             2
I_Simpson := 1/6 (b - a) (2 c[2] m a b + 3/2 c[3] a m b  + 3/2 c[3] a  m b

                 2             2             3                3
     + 2 c[2] m b  + 2 c[2] m a  + 3/2 c[3] a  m + 3/2 m c[3] b  + 3 c[1] m b

     + 3 c[1] m a + 6 c[0] m)/m
```

Les deux formules sont identiques. La méthode de Simpson est donc exacte pour les polynômes de degré 3.

```
> expand (I_exacte - I_Simpson);
                                        0
```

Théorème 2 *Fixons une formule de quadrature d'ordre p, à s étages. Supposons que la fonction $f(x)$ soit p fois dérivable sur $[a,b]$ et notons*

$$M = \sup_{x\in[a,b]} |f^{(p)}(x)| .$$

Supposons l'intervalle $[a,b]$ subdivisé en $n = (s-1)\,m$ petits intervalles de longueur h. Notons err *l'erreur commise : la différence entre l'intégrale (2.1) et la somme des m quadratures effectuées. Alors il existe une constante C indépendante de la fonction $f(x)$ et de $[a,b]$ telle que*

$$|\mathrm{err}| \quad \leqslant \quad C\,M\,(b-a)\,h^p .$$

Preuve. Voir [2, Théorème 2.3, page 8]. □

Remarque. Si $f(x)$ est un polynôme de degré strictement inférieur à p, alors sa dérivée p-ième, $f^{(p)}(x)$, est identiquement nulle et $M = 0$. D'après le théorème, $err = 0$. On retrouve la définition d'une formule de quadrature d'ordre p.

Exemple. On calcule différentes approximations de la même intégrale, en utilisant toujours la même formule de quadrature mais en multipliant par deux, à chaque ligne, le nombre d'intervalles. À chaque ligne, la longueur du pas, h, est donc divisée par deux. On affiche le nombre de bits exacts de l'estimation : l'opposé du logarithme en base deux de l'erreur relative effectuée (voir [1, section 1.3, page 9]).

```
NBSTEPS   NB BITS EXACTS APPROX.
      2   4.9250104298085873
      4   9.7804792623249970
      8   13.961492654032389
     16   18.005495742660596
     32   22.016427397654574
     64   26.019156174906570
    128   30.019838002143089
    256   34.019986935697041
    512   38.020609052052940
   1024   42.013910111798765
```

Ce nombre de bits semble augmenter de 4 à chaque ligne, ce qui signifie que l'erreur *err* du théorème 2 est divisée par 2^4 à chaque ligne. Les réels C, M et $b-a$ ne changent pas. En traitant le symbole $\leqslant$ comme une égalité, on déduit que

$$\left(\frac{h}{2}\right)^p = \frac{h^p}{2^p} = \frac{h^p}{2^4}$$

et donc que l'ordre de la formule de quadrature est $p = 4$. On a donc utilisé la formule de Simpson ou celle de Newton.

Bibliographie

[1] François Boulier. Calcul Numérique. Algèbre Linéaire. Technical report, Université Lille 1, 59655, Villeneuve d'Ascq, France, 2013. Support de cours de GIS3, accessible sur `http://cristal.univ-lille.fr/~boulier`.

[2] Ernst Hairer and Gerhard Wanner. Introduction à l'Analyse Numérique. Accessible sur `http://www.unige.ch/~hairer`, juin 2005.

Chapitre 3

Les Splines Lissantes

Ce chapitre est de nature plus technique que les autres. On y approfondit la notion de *spline lissante* — *smoothing spline* en Anglais en n'utilisant que des arguments mathématiques accessibles à un étudiant de GIS. Il s'agit d'une notion importante en *data science*, abordée dans [2] et implantée dans le logiciel R.

Le texte de référence est celui de Reinsch [6]. Ce chapitre a beaucoup bénéficié du livre de Prenter [5] qui constitue un texte de référence très lisible sur les splines en général, de l'article de Pollock [4] pour la présentation calculatoire et de la relecture du texte de Reinsch par Cline [1].

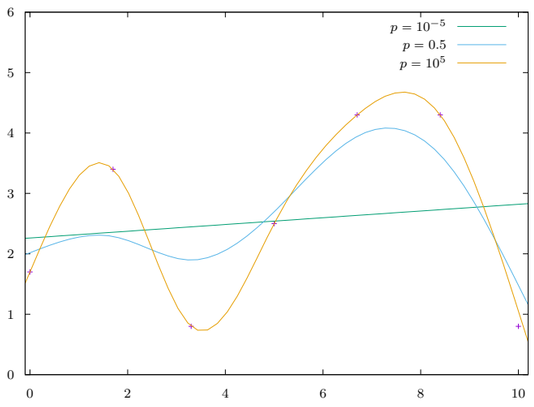

FIGURE 3.1 – Des données expérimentales et les splines lissantes pour trois valeurs du paramètre p (qui peut se déduire de S). On observe en particulier que, quand p tend vers zéro, la spline tend vers la droite de régression linéaire et que, quand p tend vers $+\infty$, elle tend vers la spline interpolante.

3.1 Le problème posé par Reinsch

On considère $n+1$ points (x_i, y_i) pour $0 \leqslant i \leqslant n$. À chaque point est associé un réel $\sigma_i > 0$. On suppose que les abscisses $a = x_0 < x_1 < \cdots < x_n = b$ (appelées aussi les *nœuds* — *knots* en Anglais) subdivisent l'intervalle $[a, b]$ en n sous-intervalles.

Reinsch cherche une fonction [1] $f(x)$ deux fois dérivable sur $[a, b]$ qui minimise

$$\int_a^b (g'')^2 \, \mathrm{d}x \tag{3.1}$$

parmi les fonctions $g(x)$ qui vérifient la contrainte (3.2) où $S \geqslant 0$ est un paramètre donné :

$$\sum_{i=0}^{n} \left(\frac{g(x_i) - y_i}{\sigma_i} \right)^2 \quad \leqslant \quad S \, . \tag{3.2}$$

3.2 La solution est une spline cubique

Il est utile de donner une définition précise des splines cubiques.

Définition 3 *Une* spline cubique *est une fonction $s(x)$ deux fois dérivable sur l'intervalle $[a, b]$, qui se réduit à un polynôme de degré 3 sur chaque sous-intervalle $[x_i, x_{i+1}]$ de $[a, b]$.*

Concrètement, toute spline cubique est donc une fonction définie par morceaux

$$s(x) = \begin{cases} s_0(x) & (x < x_1) \, , \\ s_1(x) & (x < x_2) \, , \\ \vdots & \vdots \\ s_{n-2}(x) & (x < x_{n-1}) \, , \\ s_{n-1}(x) & (\text{sinon}) \, , \end{cases}$$

où chaque morceau $s_i(x)$ est défini par un polynôme de degré 3.

Définition 4 *On dit qu'une spline cubique $s(x)$* interpole *une fonction $f(x)$ dérivable sur $[a, b]$ si $s(x_i) = f(x_i)$ pour $0 \leqslant i \leqslant n$.*

Définition 5 *Une spline cubique $s(x)$ est dite* naturelle *si $s''(a) = s''(b) = 0$.*

La proposition suivante correspond à la *first integral relation* de [5, page 96].

Proposition 1 *Soient $f(x)$ une fonction deux fois dérivable sur l'intervalle $[a, b]$ et $s(x)$ une spline cubique naturelle interpolant f. Alors*

$$\int_a^b (f'')^2 \, \mathrm{d}x \quad = \quad \int_a^b (s'')^2 \, \mathrm{d}x + \int_a^b (f'' - s'')^2 \, \mathrm{d}x \, . \tag{3.3}$$

1. Par la suite, on écrit souvent f, s ou g au lieu de $f(x), s(x)$ ou $g(x)$ pour alléger les notations.

Preuve. Comme

$$(f'')^2 \quad = \quad ((f''-s'')+s'')^2 \quad = \quad (f''-s'')^2 + 2\,(f''-s'')\,s'' + (s'')^2\,,$$

il suffit de prouver que

$$\int_a^b (f''-s'')\,s''\,\mathrm{d}x \quad = \quad 0\,.$$

Le reste de la preuve est un exercice d'intégration par parties. Comme la dérivée troisième s''' est constante sur chaque sous-intervalle, on découpe l'intégrale en

$$\int_a^b (f''-s'')\,s''\,\mathrm{d}x \quad = \quad \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} (f''-s'')\,s''\,\mathrm{d}x\,.$$

Sur chaque sous-intervalle, on a, en intégrant par parties,

$$\int_{x_{i-1}}^{x_i} (f''-s'')\,s''\,\mathrm{d}x \quad = \quad \left[(f'-s')\,s''\right]_{x_{i-1}}^{x_i} - \int_{x_{i-1}}^{x_i} (f'-s')\,s'''\,\mathrm{d}x\,.$$

Mais la somme suivante est nulle : l'égalité (1) est due à une simplification téléscopique ; l'égalité (2) est due [2] à l'hypothèse de spline naturelle : $s''(a) = s''(b) = 0$:

$$\sum_{i=1}^{n} \left[(f'-s')\,s''\right]_{x_{i-1}}^{x_i} \quad \underset{(1)}{=} \quad \left[(f'-s')\,s''\right]_a^b \quad \underset{(2)}{=} \quad 0\,.$$

Il ne reste plus qu'à vérifier que la somme suivante est nulle : l'égalité (3) s'établit en remarquant que s''' est une constante c_i sur chaque sous-intervalle $[x_{i-1}, x_i]$; l'égalité (4) en remarquant que $f - s$ est une primitive de $f' - s'$; l'égalité (5) en utilisant le fait que s interpole f.

$$\sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} (f'-s')\,s'''\,\mathrm{d}x \quad \underset{(3)}{=} \quad \sum_{i=1}^{n} c_i \int_{x_{i-1}}^{x_i} (f'-s')\,\mathrm{d}x \quad \underset{(4)}{=} \quad \sum_{i=1}^{n} c_i\,\left[f-s\right]_{x_{i-1}}^{x_i} \quad \underset{(5)}{=} \quad 0\,.$$

□

Les deux propositions suivantes sont très proches des deux théorèmes donnés dans [5, pages 100-101]. Elles découlent de la proposition 1.

Proposition 2 *La fonction identiquement nulle est l'unique spline interpolante naturelle de la fonction identiquement nulle.*

Preuve. La fonction identiquement nulle est une spline interpolante naturelle de la fonction identiquement nulle $f(x)$. Montrons l'unicité. Soit $s(x)$ une spline interpolante naturelle de $f(x)$. Alors, d'après la proposition 1 on a

$$\int_a^b (s'')^2\,\mathrm{d}x \quad \leqslant \quad \int_a^b (f'')^2\,\mathrm{d}x \quad = \quad 0\,.$$

2. Prenter suppose $f'(a) = s'(a)$ et $f'(b) = s'(b)$, ce qui conduit à la même conclusion.

Par conséquent, $s'' = 0$ sur $[a, b]$ et s est donc un polynôme de degré 1 sur $[a, b]$ qu'on peut écrire

$$s(x) \quad = \quad s(a) + s'(a)\,(x - a)\,.$$

Mais comme s interpole la fonction nulle on a $s(a) = 0$ et donc $s'(a)\,(b - a) = 0$, qui implique $s'(a) = 0$ aussi. Par conséquent s est identiquement nulle. □

Proposition 3 *Soit f une fonction deux fois dérivable sur $[a, b]$. Il existe une unique spline naturelle s qui interpole f.*

Preuve. On commence par montrer que les $4\,n$ coefficients des n cubiques qui constituent s sont contraints par un système de $4\,n$ équations linéaires. Les conditions de spline naturelle $s''(a) = s''(b) = 0$ donnent deux équations. Les conditions $s(x_i) = f(x_i)$ donnent $n + 1$ équations. Les conditions qui expriment le fait que les cubiques doivent se raccorder ainsi que leurs dérivées première et deuxième, sur les nœuds intérieurs donnent $3\,(n - 1)$ équations. Au total, on obtient $4\,n$ équations et donc un système d'équations linéaires $A\,c = v$ où A est une matrice $(4\,n) \times (4\,n)$, c est le vecteur des coefficients des cubiques et v est un vecteur ne comportant que des zéros, les valeurs de f aux nœuds et les valeurs de f' en a et en b. La proposition 2 montre que dans le cas où f est la fonction identiquement nulle, ce système admet une unique solution. Donc la matrice A est inversible. Donc la spline s existe et est unique quelle que soit la fonction f. □

La proposition suivante aussi est une conséquence de la proposition 1. Elle est plus ou moins écrite dans [5, page 101].

Proposition 4 *Soit $f(x)$ est une fonction quelconque, deux fois dérivable sur $[a, b]$, interpolant $n + 1$ points d'abscisses $a = x_0 < x_1 < \cdots < x_n = b$, d'ordonnées $y_0, y_1, \ldots, y_n$. Si $s(x)$ est la spline cubique naturelle interpolant ces mêmes points alors*

$$\int_a^b (f'')^2 \,\mathrm{d}x \quad \geqslant \quad \int_a^b (s'')^2 \,\mathrm{d}x\,. \tag{3.4}$$

En particulier, si l'inégalité (3.4) *est une égalité alors f est une spline cubique (mais pas forcément la spline cubique naturelle).*

Preuve. La proposition 3 montre l'existence de s. L'inégalité (3.4) est une conséquence de la proposition 1 et du fait que l'intégrale (3.5) est positive ou nulle.

$$\int_a^b (f'' - s'')^2 \,\mathrm{d}x \tag{3.5}$$

Supposons que l'inégalité (3.4) soit une égalité et donc que (3.5) soit nulle. Comme f et s sont deux fois dérivables, leur dérivée seconde est continue et la différence $f'' - s''$ est la fonction nulle sur $[a, b]$. Par conséquent, f'' est un polynôme de degré 1 sur chaque sous-intervalle ; donc f est un polynôme de degré 3 sur chaque sous-intervalle ; donc f est une spline cubique d'après la définition 3. □

Corollaire 1 *Le problème posé par Reinsch a une solution quel que soit S. Cette solution est une spline cubique.*

Preuve. L'ensemble des fonctions g satisfaisant la contrainte (3.2) n'est jamais vide (même pour $S = 0$), puisqu'il contient au minimum le polynôme interpolant les $n + 1$ points. La fonction f qui minimise (3.1) existe donc toujours. D'après la proposition 4, cette fonction est une spline cubique. □

3.3 Une propriété de la dérivée troisième de la solution

Si $s(x)$ est une spline cubique, la dérivée troisième s''' n'est pas continue en général au niveau des nœuds. Pour tout nœud x_i, on distingue donc la valeur de s''' à gauche de x_i, notée $s'''(x_i)_-$, de sa valeur à droite, notée $s'''(x_i)_+$. Pour homogénéiser les notations, on pose $s'''(x_0)_- = s'''(x_n)_+ = 0$.

Soit $f(x)$ la spline cubique qui minimise (3.1) parmi les fonctions satisfaisant la contrainte (3.2). Reinsch montre qu'il existe un paramètre p tel que

$$f'''(x_i)_- - f'''(x_i)_+ \quad = \quad 2\,p\,\frac{f(x_i) - y_i}{\sigma_i^2} \qquad (0 \leqslant i \leqslant n)\,. \tag{3.6}$$

Cette section est destinée à montrer cette propriété. Dans le texte de Reinsch, p est un *multiplicateur de Lagrange*. Dans l'analyse ci-dessous, qui est due à Cline [1], on évite toute référence à cette notion.

Cline commence par démontrer la proposition suivante [1, Lemma 1], qui est une variation sur l'argument central de la proposition 1.

Proposition 5 *Soient $s(x)$ une spline cubique naturelle de nœuds $a = x_0 < x_1 < \cdots < x_n = b$ et $\ell(x)$ une fonction deux fois dérivable sur $[a, b]$. Alors*

$$\int_a^b \ell''\, s''\, \mathrm{d}x \quad = \quad \sum_{i=0}^{n} \ell(x_i)\,(s'''(x_i)_+ - s'''(x_i)_-)\,. \tag{3.7}$$

Preuve. Comme la dérivée troisième s''' est constante sur chaque sous-intervalle, on découpe l'intégrale en

$$\int_a^b \ell''\, s''\, \mathrm{d}x \quad = \quad \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} \ell''\, s''\, \mathrm{d}x\,.$$

Sur chaque sous-intervalle, on a, en intégrant par parties,

$$\int_{x_{i-1}}^{x_i} \ell''\, s''\, \mathrm{d}x \quad = \quad \big[\ell'\, s''\big]_{x_{i-1}}^{x_i} - \int_{x_{i-1}}^{x_i} \ell'\, s'''\, \mathrm{d}x$$

Mais la somme suivante est nulle : l'égalité (1) est due à une simplification téléscopique ; l'égalité (2) est due à l'hypothèse de spline naturelle :

$$\sum_{i=1}^{n} \big[\ell'\, s''\big]_{x_{i-1}}^{x_i} \quad \underset{(1)}{=} \quad \big[\ell'\, s''\big]_a^b \quad \underset{(2)}{=} \quad 0\,.$$

À partir d'ici, la preuve diffère de celle de la proposition 1. Il ne reste plus qu'à simplifier la somme suivante : l'égalité (3) s'établit en remarquant que s''' est constante et égale à $s'''(x_{i-1})_+$ sur chaque sous-intervalle $[x_{i-1}, x_i]$ et que ℓ est une primitive de ℓ' ; l'égalité (4) s'obtient en découpant la somme en deux sous-sommes ; l'égalité (5) s'obtient en décalant de 1 les indices de la première sous-somme de façon à faire apparaître $\ell(x_i)$ en facteur dans les deux expressions ; l'égalité (6) vient du fait que $s'''(x_i)_- = s'''(x_{i-1})_+$; l'égalité (7) n'est due qu'à un réarrangement des sommes ; l'égalité (8) s'obtient en utilisant le fait que $s'''(x_0)_- = s'''(x_n)_+ = 0$.

$$
\begin{aligned}
-\sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} \ell'\, s'''\, \mathrm{d}x \quad &\underset{(3)}{=} \quad -\sum_{i=0}^{n-1} s'''(x_i)_+ \,(\ell(x_{i+1}) - \ell(x_i)) \\
&\underset{(4)}{=} \quad -\sum_{i=0}^{n-1} s'''(x_i)_+ \,\ell(x_{i+1}) + \sum_{i=0}^{n-1} s'''(x_i)_+ \,\ell(x_i) \\
&\underset{(5)}{=} \quad -\sum_{i=1}^{n} s'''(x_{i-1})_+ \,\ell(x_i) + \sum_{i=0}^{n-1} s'''(x_i)_+ \,\ell(x_i) \\
&\underset{(6)}{=} \quad -\sum_{i=1}^{n} s'''(x_i)_- \,\ell(x_i) + \sum_{i=0}^{n-1} s'''(x_i)_+ \,\ell(x_i) \\
&\underset{(7)}{=} \quad s'''(x_0)_+ \,\ell(x_0) - s'''(x_n)_- \,\ell(x_n) + \sum_{i=1}^{n-1} \ell(x_i)\,(s'''(x_i)_+ - s'''(x_i)_-) \\
&\underset{(8)}{=} \quad \sum_{i=0}^{n-1} \ell(x_i)\,(s'''(x_i)_+ - s'''(x_i)_-)\,.
\end{aligned}
$$

□

La proposition suivante est [1, Lemma 2].

Proposition 6 *Soient u et v deux vecteurs non nuls de $\mathbb{R}^{n+1}$. Si, pour tout $p \in \mathbb{R}$ on a $u \neq p\,v$ alors il existe un vecteur w tel que $w^T\,u > 0$ et $w^T\,v < 0$.*

Preuve. On note $\|\cdot\|$ la norme euclidienne. Supposons que pour tout $p \in \mathbb{R}$ on ait $u \neq p\,v$. Les deux vecteurs ne sont donc pas colinéaires et, d'après l'inégalité de Cauchy-Schwartz on a $u^T\,v < \|u\|\,\|v\|$. Par conséquent,

$$\|u\| - \frac{u^T\,v}{\|v\|} > 0 \quad \text{et} \quad \|v\| - \frac{u^T\,v}{\|u\|} > 0\,.$$

Le vecteur w ci-dessous vérifie $w^T\,u > 0$ et $w^T\,v < 0$:

$$w \quad = \quad \frac{u}{\|u\|} - \frac{v}{\|v\|}.$$

□

Proposition 7 *Supposons $S > 0$. Soit $s(x)$ une spline cubique naturelle de nœuds $a = x_0 < x_1 < \cdots < x_n = b$ et vérifiant la contrainte (3.2). Si, pour tout $p \in \mathbb{R}$ on a*

$$s'''(x_i)_- - s'''(x_i)_+ \quad \neq \quad 2\,p\,\frac{s(x_i) - y_i}{\sigma_i^2} \qquad (0 \leqslant i \leqslant n)\,. \tag{3.8}$$

alors il existe une autre fonction $t(x)$ deux fois dérivable sur $[a,b]$, satisfaisant la contrainte (3.2) *et telle que*

$$\int_a^b (s'')^2 \,\mathrm{d}x \quad > \quad \int_a^b (t'')^2 \,\mathrm{d}x \,. \tag{3.9}$$

En particulier, s n'est pas une solution du problème de Reinsch.

Preuve. La non existence de p implique que l'une des quantités $s'''(x_i)_- - s'''(x_i)_+$ soit non nulle. On distingue deux cas. Premier cas : on suppose que l'une au moins des quantités $s(x_i) - y_i$ est non nulle. Alors, d'après la proposition 6, il existe un vecteur w tel que

$$\sum_{i=0}^{n} w_i \,(s'''(x_i)_- - s'''(x_i)_+) > 0 \quad \text{et} \quad \sum_{i=0}^{n} w_i \,\frac{s(x_i) - y_i}{\sigma_i^2} < 0 \,.$$

Soit $\ell(x)$ une fonction deux fois dérivable sur $[a,b]$ telle que $\ell(x_i) = w_i$ pour $0 \leqslant i \leqslant n$ (le polynôme d'interpolation convient). Alors, en remplaçant w_i par $\ell(x_i)$ dans les inégalités précédentes, on obtient :

$$\sum_{i=0}^{n} \ell(x_i) \,(s'''(x_i)_- - s'''(x_i)_+) \quad > \quad 0 \,, \tag{3.10}$$

$$\sum_{i=0}^{n} \ell(x_i) \,\frac{s(x_i) - y_i}{\sigma_i^2} \quad < \quad 0 \,. \tag{3.11}$$

Pour tout $\mu \in \mathbb{R}$ on a les égalités suivantes : l'égalité (1) s'obtient en développant le carré ; l'égalité (2) s'obtient en appliquant la proposition 5 sur l'intégrale du milieu :

$$\begin{aligned}
\int_a^b (s'' + \mu\,\ell'')^2 \,\mathrm{d}x \quad &\underset{(1)}{=} \quad \int_a^b (s'')^2 \,\mathrm{d}x + 2\,\mu \int_a^b \ell''\, s'' \,\mathrm{d}x + \mu^2 \int_a^b (\ell'')^2 \,\mathrm{d}x \\
&\underset{(2)}{=} \quad \int_a^b (s'')^2 \,\mathrm{d}x \;\underbrace{+ 2\,\mu \sum_{i=0}^{n} \ell(x_i) \,(s'''(x_i)_+ - s'''(x_i)_-) + \mu^2 \int_a^b (\ell'')^2 \,\mathrm{d}x}_{\text{négatif pour } \mu > 0 \text{ très petit}} \,.
\end{aligned}$$

D'après l'inégalité (3.10), le terme du milieu est négatif pour tout $\mu > 0$ et, pour peu que ce même μ soit suffisamment petit, la somme des deux derniers termes est négative. La fonction $t = s + \mu\,\ell$ vérifie l'inégalité (3.9). Pour montrer que s n'est pas la solution optimale (pour le premier cas), il ne reste plus qu'à montrer que t vérifie la contrainte (3.2). On a l'égalité suivante :

$$\begin{aligned}
&\sum_{i=0}^{n} \left(\frac{s(x_i) + \mu\,\ell(x_i) - y_i}{\sigma_i^2} \right)^2 \quad = \\
&\qquad \sum_{i=0}^{n} \left(\frac{s(x_i) - y_i}{\sigma_i^2} \right)^2 \underbrace{+ 2\,\mu \sum_{i=0}^{n} \ell(x_i)\, \frac{s(x_i) - y_i}{\sigma_i^2} + \mu^2 \sum_{i=0}^{n} \left(\frac{\ell(x_i)}{\sigma_i^2} \right)^2}_{\text{négatif pour } \mu > 0 \text{ très petit}} \,.
\end{aligned} \tag{3.12}$$

D'après l'inégalité (3.11), le terme du milieu est négatif pour tout $\mu > 0$ et, pour peu que ce même μ soit suffisamment petit, la somme des deux derniers termes est négative. Par conséquent, si s vérifie la contrainte (3.2) alors la fonction $t = s + \mu\,\ell$ la vérifie également, ce qui achève la preuve du premier cas.

Second cas : on suppose que toutes les quantités $s(x_i) - y_i$ sont nulles. Il est encore possible de trouver un vecteur w tel que l'inégalité (3.10) soit satisfaite (il suffit de prendre des w_i de même signe que les $s'''(x_i)_- - s'''(x_i)_+$). Par conséquent, en construisant $\ell(x)$ comme dans le premier cas et en prenant $\mu > 0$ suffisamment petit, on obtient une fonction $t = s + \mu\,\ell$ qui vérifie (3.9). Pour montrer que s n'est pas la solution optimale (pour le second cas), il ne reste plus qu'à montrer que t vérifie la contrainte (3.2). On a l'égalité suivante qui est une forme simplifiée de (3.12) :

$$\sum_{i=0}^{n} \left(\frac{s(x_i) + \mu\,\ell(x_i) - y_i}{\sigma_i^2} \right)^2 \quad = \quad \mu^2 \sum_{i=0}^{n} \left(\frac{\ell(x_i)}{\sigma_i^2} \right)^2$$

Comme on a supposé $S > 0$, en prenant μ suffisamment petit, cette expression est inférieure ou égale à S. La fonction $t = s + \mu\,\ell$ vérifie donc la contrainte (3.2) et la proposition est prouvée dans tous les cas. □

3.4 Calcul des coefficients de la spline

Dans cette section-ci, on revient aux explications données par Reinsch. On obtient des formules pour les coefficients de la spline qui utilisent la relation (3.6) et dépendent donc du paramètre $p > 0$ (le *multiplicateur de Lagrange*) introduit dans cette relation. On verra en section 3.5 comment calculer p à partir de S et réciproquement.

On note $f(x)$ la spline cubique naturelle solution du problème de Reinsch. Cette spline est composée de polynômes de degré 3 :

$$f_i(x) \quad = \quad a_i + b_i\,(x - x_i) + c_i\,(x - x_i)^2 + d_i\,(x - x_i)^3\,, \quad (x_i \leqslant x \leqslant x_{i+1})\,.$$

Écrire chaque polynôme f_i sous la forme d'un développement limité en $x = x_i$ simplifie considérablement les formules qui donnent leurs coefficients. Notons

$$h_i \quad = \quad x_{i+1} - x_i\,, \quad (0 \leqslant i \leqslant n-1)\,.$$

La continuité de la dérivée seconde de f aux nœuds intérieurs se traduit par le fait que $f''_{i+1}(x_{i+1}) = 2\,c_{i+1}$ doit être égal à $f''_i(x_{i+1}) = 6\,d_i\,h_i + 2\,c_i$. En tirant d_i on obtient :

$$d_i \quad = \quad \frac{c_{i+1} - c_i}{3\,h_i}\,, \quad (0 \leqslant i \leqslant n-1)\,. \tag{3.13}$$

La continuité de f aux nœuds intérieurs se traduit par le fait que $f_{i+1}(x_{i+1}) = a_{i+1}$ doit être égal à $f_i(x_{i+1}) = a_i + b_i\,h_i + c_i\,h_i^2 + d_i\,h_i^3$. En tirant b_i on obtient :

$$b_i \quad = \quad \frac{a_{i+1} - a_i}{h_i} - c_i\,h_i - d_i\,h_i^2\,, \quad (0 \leqslant i \leqslant n-1)\,. \tag{3.14}$$

La continuité de la dérivée première de f aux nœuds intérieurs se traduit par le fait que $f_i'(x_i) = b_i$ doit être égal à $f_{i-1}'(x_i) = b_{i-1} + 2\,c_{i-1}\,h_{i-1} + 3\,d_{i-1}\,h_{i-1}^2$. En remplaçant b_{i-1}, d_{i-1}, b_i et d_i par leur valeur (3.13), (3.14), on obtient

$$\frac{h_{i-1}}{3}\,c_{i-1} + 2\left(\frac{h_{i-1}}{3} + \frac{h_i}{3}\right)c_i + \frac{h_i}{3}\,c_{i+1} \quad = \quad \frac{1}{h_{i-1}}\,a_{i-1} - \left(\frac{1}{h_{i-1}} + \frac{1}{h_i}\right)a_i + \frac{1}{h_i}\,a_{i+1}\,, \quad (1 \leqslant i \leqslant n-1)\,. \tag{3.15}$$

Enfin, la relation (3.6) se traduit par le fait que $f_i'''(x_i) - f_{i-1}'''(x_i) = 6\,d_i - 6\,d_{i-1}$ doit être égal à $2\,p\,(y_i - f_i(x_i))/\sigma_i^2$. En remplaçant d_i et d_{i-1} par leur valeur (3.13) on obtient

$$\frac{1}{h_{i-1}}\,c_{i-1} - \left(\frac{1}{h_{i-1}} + \frac{1}{h_i}\right)c_i + \frac{1}{h_i}\,c_{i+1} \quad = \quad p\,\frac{y_i - a_i}{\sigma_i^2}\,, \quad (1 \leqslant i \leqslant n-1)\,. \tag{3.16}$$

3.4.1 Formulation matricielle

Introduisons la matrice T de dimension $(n-1) \times (n-1)$ suivante (attention au coefficient 1/3)

$$T \;=\; \frac{1}{3}\begin{pmatrix} 2\,(h_0+h_1) & h_1 & 0 & \dots & 0 & 0 \\ h_1 & 2\,(h_1+h_2) & h_2 & \dots & 0 & 0 \\ 0 & h_2 & 2\,(h_2+h_3) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 2\,(h_{n-3}+h_{n-2}) & h_{n-2} \\ 0 & 0 & 0 & \dots & h_{n-2} & 2\,(h_{n-2}+h_{n-1}) \end{pmatrix},$$

la matrice Q de dimension $(n+1) \times (n-1)$ suivante où $g_i = 1/h_i$:

$$Q \;=\; \begin{pmatrix} g_0 & 0 & \dots & 0 & 0 \\ -g_0-g_1 & g_1 & \dots & 0 & 0 \\ g_1 & -g_1-g_2 & \dots & 0 & 0 \\ 0 & g_2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & -g_{n-3}-g_{n-2} & g_{n-2} \\ 0 & 0 & \dots & g_{n-2} & -g_{n-2}-g_{n-1} \\ 0 & 0 & \dots & 0 & g_{n-1} \end{pmatrix},$$

la matrice diagonale Σ formée de $\sigma_0, \sigma_1, \dots, \sigma_n$, ainsi que les vecteurs $a = (a_0, a_1, \dots, a_n)^T$, $c = (c_1, c_2, \dots, c_{n-1})^T$ et $y = (y_0, y_1, \dots, y_n)^T$. Les relations (3.13) et (3.16) auxquelles il faut ajouter les conditions de spline naturelle $c_0 = c_n = 0$ s'écrivent alors

$$T\,c \;=\; Q^T\,a\,, \tag{3.17}$$

$$Q\,c \;=\; p\,\Sigma^{-2}\,(y - a)\,. \tag{3.18}$$

En multipliant (3.18) à gauche par $Q^T\,\Sigma^2$ et en éliminant $Q^T\,a$ grâce à (3.17) on trouve une équation matricielle qui ne dépend que du vecteur c :

$$(Q^T\,\Sigma^2\,Q + p\,T)\,c \quad = \quad p\,Q^T\,y \tag{3.19}$$

ainsi qu'une équation pour le vecteur a :

$$a \quad = \quad y - p^{-1}\,\Sigma^2\,Q\,c\,. \tag{3.20}$$

La matrice des coefficients de l'équation (3.19) est symétrique à cinq bandes. Elle est définie positive si $p > 0$. Il ne reste plus qu'à résoudre (3.19) (Cholesky) pour obtenir c, reporter dans (3.20) pour obtenir a puis utiliser (3.13) et (3.14) pour obtenir d et b.

3.5 Calcul du paramètre de lissage

Pour comparer le texte fondateur de Reinsch avec des travaux plus modernes [2], il est utile de présenter la formulation du problème de Reinsch par la formule (3.22). On en profite pour montrer que la borne S est atteinte à l'optimum (ce qui peut aussi se montrer sans la formulation Lagrangienne, comme le fait remarquer Cline). Dans les sections suivantes, on montre comment calculer p à partir de S, en utilisant d'intéressantes techniques de dérivation d'expressions matricielles. Puis on introduit la notion de degré de liberté, qui fournit un paramétrage plus élégant, utilisable dans des contextes plus généraux et qui est implanté dans le logiciel R. Enfin, on montre comment définir le paramètre optimal (en un certain sens), ce qui permet de déterminer automatiquement le lissage à appliquer lorsque l'utilisateur ne souhaite pas préciser le degré de liberté qu'il souhaite.

3.5.1 Formulation avec multiplicateur de Lagrange

La méthode du *multiplicateur de Lagrange* est une méthode générale permettant de transformer un problème d'optimisation sous contrainte en un problème d'optimisation équivalent, sans contrainte, mais au prix de l'introduction d'une variable supplémentaire appelée *multiplicateur de Lagrange*.

Le problème posé par Reinsch consiste à minimiser (3.1) sous la contrainte d'inégalité (3.2). Pour appliquer la méthode du multiplicateur de Lagrange, il est nécessaire de transformer la contrainte d'inégalité en une contrainte d'égalité. On introduit une première variable z (une variable d'écart) pour représenter la différence entre les deux membres de l'inégalité (3.2) et on code le sens de l'inégalité en posant que cette différence est égale à z^2 (nécessairement positif ou nul) plutôt qu'à z :

$$\sum_{i=0}^{n} \left(\frac{g(x_i) - y_i}{\sigma_i} \right)^2 \quad = \quad S - z^2\,. \tag{3.21}$$

D'après la théorie de la méthode du multiplicateur de Lagrange, minimiser (3.1) sous la contrainte (3.21) équivaut à minimiser le critère suivant, sans contrainte, où on a introduit une seconde variable p

(le multiplicateur de Lagrange) :

$$\mathscr{L} \quad = \quad \int_a^b (g'')^2 \, \mathrm{d}x + p \left(\sum_{i=0}^{n} \left(\frac{g(x_i) - y_i}{\sigma_i} \right)^2 - S + z^2 \right) . \tag{3.22}$$

Comme dans le cas classique des fonctions de plusieurs variables réelles, toute solution optimale de (3.22) annule le système des dérivées partielles du critère, par rapport à chacune des variables dont il dépend. En particulier, toute solution optimale de (3.22) annule

$$\frac{\partial \mathscr{L}}{\partial z} \quad = \quad 2\,p\,z\,.$$

À l'optimum, on a donc soit $p = 0$ et la solution de (3.22) est une droite, soit $z = 0$ (le cas qui nous intéresse). Supposons $z = 0$. D'après (3.21), on conclut que S est égale à

$$\sum_{i=0}^{n} \left(\frac{g(x_i) - y_i}{\sigma_i} \right)^2 . \tag{3.23}$$

Avec les notations de la section 3.4.1 et les formules (3.19) et (3.20), la somme (3.23) est le carré de la norme deux d'un certain vecteur et peut s'écrire :

$$\left(\|\Sigma^{-1} \, (a - y)\|_2 \right)^2 \quad = \quad \left(\|\Sigma \, Q \, (Q^T \, \Sigma^2 \, Q + p \, T)^{-1} \, Q^T \, y\|_2 \right)^2 \quad = \quad F(p)^2 \, .$$

Le paramètre p de la formule ci-dessus est celui introduit dans la relation (3.6). Il se trouve que c'est le même paramètre que le multiplicateur de Lagrange p (voir l'article de Reinsch) mais c'est sans importance pour ce qui nous occupe. On a donc établi que

$$S \quad = \quad F(p)^2 \, . \tag{3.24}$$

3.5.2 Calcul de p en fonction de S

Pour calculer S en fonction de p, il suffit d'évaluer $F(p)$. Pour calculer p en fonction de S, Reinsch propose d'utiliser une méthode de Newton, ce qui suppose de mettre au point une formule pour la dérivée $F'(p)$. C'est ce qu'on explique dans cette section.

On définit la dérivée u' d'un vecteur dont les coordonnées sont des fonctions u_i comme le vecteur des dérivées u_i'.

Notons $u = (Q^T \, \Sigma^2 \, Q + p \, T)^{-1} \, Q^T \, y$. Le vecteur u dépend de p. Ses coordonnées sont donc des fonctions de p. Notons $v = \Sigma \, Q \, u$. D'une part, d'après la règle sur la transposée d'un produit et le fait que Σ est égale à sa transposée, on a $v^T = u^T \, Q^T \, \Sigma$; d'autre part, comme les matrices Σ et Q ne dépendent pas de p et que la dérivée d'une somme est la somme des dérivées, on voit que $v' = \Sigma \, Q \, u'$. Enfin, en développant les expressions, on voit que

$$F(p)^2 \quad = \quad v^T \, v \, . \tag{3.25}$$

Par conséquent,

$$(F(p)^2)' \quad = \quad 2 \, v^T \, v' \quad = \quad 2 \, u^T \, Q^T \, \Sigma^2 \, Q \, u' \, .$$

On cherche maintenant une formule pour u'. La définition de u implique que

$$(Q^T\,\Sigma^2\,Q + p\,T)\,u = Q^T\,y \quad \text{et donc} \tag{3.26}$$
$$Q^T\,\Sigma^2\,Q\,u + T\,(p\,u) = Q^T\,y\,. \tag{3.27}$$

Comme les matrices et vecteurs Q, T, Σ et y ne dépendent pas de p on a, en dérivant par rapport à p,

$$Q^T\,\Sigma^2\,Q\,u' = -T\,(u + p\,u') \quad \text{et donc} \tag{3.28}$$
$$u' = -(Q^T\,\Sigma^2\,Q + p\,T)^{-1}\,T\,u \tag{3.29}$$

Par conséquent,

$$\frac{1}{2}\,(F(p)^2)' = p\,u^T\,T\,(Q^T\,\Sigma^2\,Q + p\,T)^{-1}\,T\,u - u^T\,T\,u\,. \tag{3.30}$$

On a tout ce qui est nécessaire pour appliquer la méthode de Newton (voir la section 4.6.4). Dans [7, page 453], Reinsch conseille de partir de l'équation suivante, pour une convergence plus rapide :

$$\frac{1}{F(p)} = \frac{1}{\sqrt{S}} \tag{3.31}$$

3.5.3 Paramétrage par le degré de liberté

Paramétrer l'importance du lissage par p ou par S n'est pas très satisfaisant. Qui plus est, ces deux paramètres sont étroitement liés à la définition des splines lissantes telle que Reinsch l'a donnée. Dans la littérature, on présente parfois les splines comme des fonctions g qui minimisent d'autres critères comme [2, Eq (5.9), page 151] :

$$\mathrm{RSS}(g,\lambda) = \sum_{i=0}^{n}(y_i - g(x_i))^2 + \lambda\int_{x_0}^{x_n}(g''(x))^2\,\mathrm{d}x\,. \tag{3.32}$$

Le paramètre λ ressemble fort à un multiplicateur de Lagrange mais ce n'en est pas un : il n'est pas multiplié par la contrainte ! Pour autant, il joue approximativement le rôle de $1/p$ et permet de contrôler l'importance du lissage. Et il existe plein de variantes de ces définitions. Le critère (3.32) est probablement adapté de [3, Eq (1.2), page 378] où la somme est divisée par le nombre de points.

La notion de *degré de liberté* [2, section 5.4.1, page 153] a le gros avantage d'être définie à partir de la spline solution et pas du critère. L'idée consiste à exprimer le vecteur a, qui contient les ordonnées de la spline aux abscisses x_i ($0 \leqslant i \leqslant n$), comme une fonction linéaire du vecteur y des ordonnées des points initiaux. Notons donc S_p la matrice telle que

$$a = S_p\,y\,. \tag{3.33}$$

Cette matrice se définit facilement à partir des équations de la section 3.4.1 par

$$S_p = I - \Sigma^2\,Q\,(Q^T\,\Sigma^2\,Q + p\,T)^{-1}\,Q^T\,. \tag{3.34}$$

Elle est symétrique définie positive. Ses valeurs propres sont donc réelles et positives. On peut montrer que les deux plus grandes sont exactement égales à 1 [2, page 156]. L'idée consiste à définir le *degré de liberté* de la spline comme la trace de S_p :

$$\mathrm{df}_p \quad = \quad \mathrm{Tr}(S_p)\,. \tag{3.35}$$

La trace d'une matrice est la somme de ses valeurs propres. C'est aussi la somme de ses éléments diagonaux. On a donc $2 < \mathrm{df}_p \leqslant n+1$.

L'idée, c'est que le rang d'une matrice est égal au nombre de ses valeurs propres non nulles et que la trace de S_p est une mesure de la dimension de l'image de S_p et donc une mesure du nombre de paramètres effectifs dont dépend la spline.

Les splines lissantes implantées dans le logiciel R sont paramétrées par ce degré de liberté.

3.5.4 Calcul du paramètre optimal

On peut enfin vouloir déterminer le paramètre (quel qu'il soit) de lissage à partir d'un autre critère à optimiser. Une idée populaire consiste à chercher le paramètre qui minimise le critère [2, Eq. (5.26) ; page 161]

$$\mathrm{CV}(g) \quad = \quad \sum_{i=0}^{n} (y_i - g^{[i]}(x_i))^2 \tag{3.36}$$

où $g^{[i]}$ est la spline lissante obtenue en tenant compte de tous les points sauf le i-ème. Il s'agit d'une technique de *validation croisée* où on ne laisse de côté qu'un seul point (*leave one out cross validation*). Et il est remarquable que (3.36) peut être calculée à partir de la spline lissante g obtenue en tenant compte de tous les points, grâce à la formule suivante [2, Eq. (5.27) ; page 161], où on a supposé que les lignes et colonnes de S_p sont indicées à partir de 0 :

$$\mathrm{CV}(g) \quad = \quad \sum_{i=0}^{n} \left(\frac{y_i - g(x_i)}{1 - S_p(i,i)} \right)^2 . \tag{3.37}$$

C'est ce type de méthode qui permet de déterminer automatiquement le paramètre de lissage dans un logiciel comme R, lorsque l'utilisateur ne spécifie pas le degré de liberté voulu.

Bibliographie

[1] Alan Kaylor Cline. An Expansion of the Derivation of the Spline Smoothing Theory. `https://pdfs.semanticscholar.org/ae2c/118aaa18beeded133276c0cf0c9ee4c62d1d.pdf`, 2017.

[2] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning. Data Mining, Inference and Prediction*. Springer Series in Statistics. Springer, 2nd edition, 2009. Available at `https://web.stanford.edu/~hastie/Papers/ESLII.pdf`.

[3] Peter Craven and Grace Wahba. Smoothing Noisy Data with Spline Functions. *Numerische Mathematik*, 31 :377–403, 1979.

[4] D. S. G. Pollock. Smoothing with cubic splines. *Handbook of Time Series Analysis, Signal Processing, and Dynamics*, 12 1999. Available at `www.physics.muni.cz/~jancely/NM/Texty/Numerika/CubicSmoothingSpline.pdf`.

[5] Paddy M. Prenter. *Splines and Variational Methods*. Pure and Applied Mathematics. Wiley, New York, 1975.

[6] Christian H. Reinsch. Smoothing by Spline Functions. *Numerische Mathematik*, 10 :177–183, 1967.

[7] Christian H. Reinsch. Smoothing by Spline Functions II. *Numerische Mathematik*, 16 :451–454, 1971.

Chapitre 4

Intégration Numérique des Équations Différentielles

4.1 Introduction

L'objet de ce cours est l'étude des méthodes numériques d'intégration d'équations différentielles ordinaires [1]. Il existe des méthodes plus ou moins complexes, qui permettent de traiter des exemples plus ou moins difficiles. Pour les comparer, nous allons suivre un exemple issu de l'astronomie [2] [2, page 129] : le problème restreint à trois corps. On considère deux corps de masses $\mu' = 1 - \mu$ (la Terre) et μ (la Lune), en rotation circulaire dans un plan utour de leur centre de gravité, ainsi qu'un troisième corps, de masse négligeable (un satellite) se déplaçant dans le même plan, sous l'action de la gravitation. La variable indépendante x représente le temps. Les deux variables dépendantes $y_1(x)$ et $y_2(x)$ donnent les coordonnées cartésiennes du satellite dans le plan. Le système est constitué de deux équations différentielles ordinaires d'ordre deux :

$$\begin{aligned} \ddot{y}_1 &= y_1 + 2\,\dot{y}_2 - \mu' \frac{y_1 + \mu}{D_1} - \mu \frac{y_1 - \mu'}{D_2}\,, \\ \ddot{y}_2 &= y_2 - 2\,\dot{y}_1 - \mu' \frac{y_2}{D_1} - \mu \frac{y_2}{D_2}\,, \\ D_1 &= ((y_1 + \mu)^2 + y_2^2)^{\frac{3}{2}}\,, \\ D_2 &= ((y_1 - \mu')^2 + y_2^2)^{\frac{3}{2}}\,. \end{aligned} \tag{4.1}$$

1. On dit d'une équation différentielle, qu'elle est *ordinaire*, si elle ne fait figurer qu'une seule dérivation $\mathrm{d}/\mathrm{d}x$, par rapport à une seule *variable indépendante* x. Une équation où figurent des dérivations par rapport à plusieurs variables indépendantes, est dite *aux dérivées partielles*.

2. Par un heureux hasard, j'avais commencé à préparer ces notes de cours (et l'exemple principal) deux jours avant l'annonce de la rencontre de Rosetta et de la comète Churyomov-Gerasimenko ! Les équations (4.1) ne sont pas celles qui ont guidé Rosetta sur sa cible mais ce sont leurs petites sœurs. Certaines techniques mathématiques présentées dans ce cours datent du milieu des années 1980. Sachant que le principe d'une mission telle que celle de Rosetta a été entériné par l'Europe en 1985, on peut penser qu'elles ont pu être utilisées pour des études préliminaires.

Les valeurs des constantes sont

$$\mu = 0.012277471\,, \quad \mu' = 1 - \mu = 0.987722529\,.$$

Il se trouve que, pour les conditions initiales suivantes (initialement, le satellite est proche (un peu au delà) de la Lune, située en $(\mu', 0)$, avec une vecteur vitesse parallèle à l'axe vertical y_2) :

$$\begin{aligned} &y_1(0) = 0.994\,, \quad \dot{y}_1(0) = 0\,, \quad y_2(0) = 0\,, \\ &\dot{y}_2(0) = -2.00158510637908252240537862224\,, \end{aligned} \tag{4.2}$$

la solution est périodique (la courbe dans le plan (y_1, y_2) est fermée), de période

$$x_{\text{end}} = 17.06521656015796255889172062 49\,.$$

De telles solutions périodiques ont fasciné les mathématiciens pendant des dizaines d'années et sont aujourd'hui appelées orbites d'Arenstorf[3] . Voir Figure 4.1. Quant au problème des trois corps, il fascine les physiciens depuis 300 ans[4].

3. Richard Arenstorf est un mathématicien Américain, né en 1929, qui travaillait pour la NASA à l'époque de la catastrophique mission Apollo 13. L'orbite qu'il a mise au point à l'aide d'un ordinateur a servi d'orbite de secours, et permis à l'équipage de rentrer sur Terre [7]. Richard Arenstorf est mort le 18 septembre 2014.

4. Voici une mise en équation. On place l'origine $(0, 0)$ sur le centre de gravité du système Terre-Lune, qui se trouve toujours dans l'axe Terre-Lune. Par un calcul de barycentre, on trouve les coordonnées de la Terre $(-\mu, 0)$ et de la Lune $(\mu', 0)$. Graphiquement, on a envie de garder l'axe Terre-Lune horizontal. La position (y_1, y_2) du satellite est donc prise par rapport à un repère en rotation par rapport à un autre repère supposé fixe, suivant une vitesse angulaire θ. D'après [6, chapter I, pages 17-18] et l'hypothèse que la rotation est circulaire, les composantes du vecteur accélération sont :

$$\begin{pmatrix} \ddot{y}_1 - y_1\,\dot{\theta}^2 - y_2\,\ddot{\theta} - 2\,\dot{y}_2\,\dot{\theta} \\ \ddot{y}_2 - y_2\,\dot{\theta}^2 + y_1\,\ddot{\theta} + 2\,\dot{y}_1\,\dot{\theta} \end{pmatrix}.$$

On suppose pour simplifier que la vitesse angulaire est constante ($\ddot{\theta} = 0$) et même que $\dot{\theta} = 1$. Le vecteur accélération se simplifie en :

$$\begin{pmatrix} \ddot{y}_1 - y_1 - 2\,\dot{y}_2 \\ \ddot{y}_2 - y_2 + 2\,\dot{y}_1 \end{pmatrix}.$$

D'après la loi de la gravitation de Newton et, plus particulièrement, [6, chapter XIII, page 353, *The restricted problem of three bodies*], ce vecteur accélération est égal à

$$\begin{pmatrix} \frac{\partial F}{\partial y_1} \\ \frac{\partial F}{\partial y_2} \end{pmatrix} \quad \text{où} \quad F = \frac{\mu'}{((y_1+\mu)^2 + y_2^2)^{\frac{1}{2}}} + \frac{\mu}{((y_1-\mu')^2 + y_2^2)^{\frac{1}{2}}}.$$

Les numérateurs sont les deux masses de la Terre et de la Lune. Les dénominateurs sont la distance du satellite à la Terre et celle du satellite à la Lune. En développant les expressions, on trouve le système (4.1) :

$$\begin{pmatrix} \ddot{y}_1 - y_1 - 2\,\dot{y}_2 \\ \ddot{y}_2 - y_2 + 2\,\dot{y}_1 \end{pmatrix} = \begin{pmatrix} -\mu' \dfrac{y_1+\mu}{((y_1+\mu)^2 + y_2^2)^{\frac{3}{2}}} - \mu \dfrac{y_1-\mu'}{((y_1-\mu')^2 + y_2^2)^{\frac{3}{2}}} \\ -\mu' \dfrac{y_2}{((y_1+\mu)^2 + y_2^2)^{\frac{3}{2}}} - \mu \dfrac{y_2}{((y_1-\mu')^2 + y_2^2)^{\frac{3}{2}}} \end{pmatrix}.$$

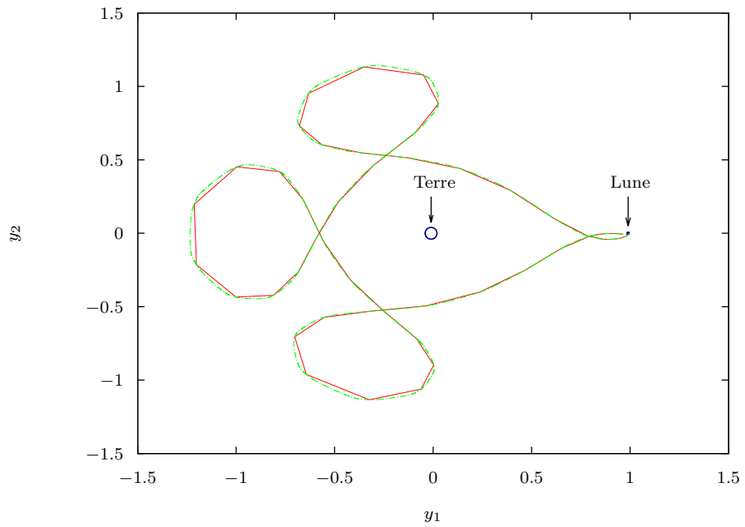

FIGURE 4.1 – Une orbite d'Arenstorf calculée par un intégrateur de type Runge-Kutta à pas adaptatif. La variable y_1 est en abscisse ; y_2 est en ordonnée. L'intégrateur, élaboré autour du schéma Dormand-Prince 5(4) , n'a calculé que la soixantaine de sommets de la courbe polygonale. L'erreur finale (sur l'exemple, il s'agit de la distance entre le point de départ et le point d'arrivée, puisqu'en théorie, la courbe est fermée) est de l'ordre de 10^{-2}. La courbe lisse a été obtenue en évaluant des polynômes d'interpolation entre les sommets par une méthode, dite de sortie dense . On remarque que les raccords entre les morceaux de courbes, au niveau des sommets, sont lisses. L'intégrateur est dit à pas adaptatif parce qu'il effectue des petits pas dans les virages serrés et des pas plus grands ailleurs : les segments de la courbe polygonale ne sont pas tous de même longueur.

4.2 Éléments théoriques

Une équation différentielle en une variable est définie par une fonction

$$f : [x_0,\, x_{\text{end}}] \times \mathbb{R} \to \mathbb{R}\,, \quad (x,\, y) \mapsto f(x,\, y)\,.$$

L'équation définie par la fonction f est alors :

$$\frac{\mathrm{d}y}{\mathrm{d}x}\,(x) = f(x,\, y(x))\,, \quad \text{notée aussi} \quad \dot{y} = f(x, y)\,.$$

La solution d'une équation différentielle est, en général, une famille de fonctions de x. Pour préciser l'une des fonctions de la famille, il suffit, en général, de fixer une *condition initiale* $y(x_0) = \alpha$. La donnée d'une équation différentielle plus une condition initiale s'appelle un *problème de Cauchy* en Français et un initial value problem en Anglais.

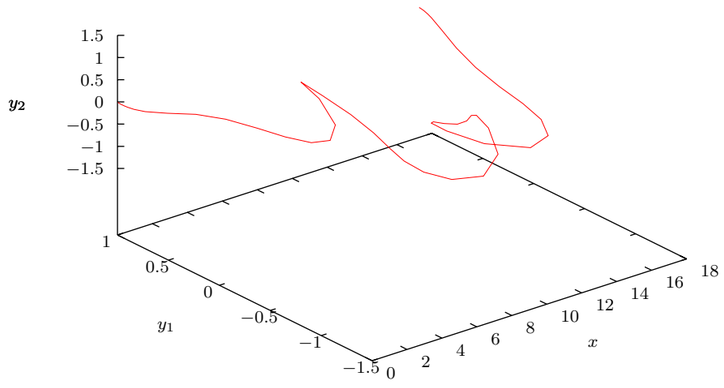

FIGURE 4.2 – La même orbite d'Arenstorf, mais représentée en trois dimensions, avec l'axe des x (le temps). La courbe de la Figure 4.1 est la projection de cette courbe-ci dans le plan (y_1, y_2).

Premier exemple. Par exemple, la fonction

$$f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}\,, \quad (x, y) \mapsto x + y$$

définit l'équation différentielle

$$\dot{y} = x + y\,.$$

Si on lui ajoute la condition initiale $y(0) = \alpha$, on obtient la solution suivante (on voit que la solution dépend du paramètre α) :

$$y(x) = e^x\,(1 + \alpha) - x - 1\,.$$

L'équation logistique. La fonction suivante

$$f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}\,, \quad (x, y) \mapsto y\,(1 - y)$$

définit l'équation différentielle

$$\dot{y} = y\,(1 - y)\,.$$

La fonction f ne dépend pas de x : on dit que l'équation différentielle est *autonome*. Elle est souvent utilisée pour modéliser l'évolution d'une population $y(x)$ au cours du temps, en présence de ressources limitées. L'équation a deux solutions constantes $y(x) = 0$ (espèce éteinte) et $y(x) = 1$

(population maximale, compte tenu des ressources), obtenues avec les conditions initiales $\alpha = 0$ et $\alpha = 1$. Si y est compris entre 0 et 1, on voit que $f(x, y) > 0$, ce qui implique que $y(x)$ est croissante (si les ressources sont suffisantes, la population augmente). Si $y > 1$, on a $f(x, y) < 0$, ce qui implique que la population est décroissante (si les ressources sont insuffisantes, la population diminue). La solution exacte est

$$y(x) = -\frac{\alpha}{e^x\,(\alpha - 1) - \alpha}\cdot$$

Explosion en temps fini. Dans l'exemple précédent la solution $y(x)$ est définie pour tout $x \in \mathbb{R}$ mais ce n'est pas toujours le cas. Prenons par exemple, la fonction [5, Example 1.2]

$$f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}\,, \quad (x, y) \mapsto y^2\,,$$

qui définit l'équation différentielle

$$\dot{y} = y^2\,.$$

Pour la condition initiale $y(0) = \alpha$, la fonction suivante est une solution :

$$y(x) = \frac{\alpha}{1 - x\,\alpha}\cdot$$

Supposons $\alpha > 0$. La solution $y(x)$ vaut $+\infty$ en $x = 1/\alpha$. La borne x_{end} ne peut donc pas dépasser la valeur limite $1/\alpha$, qui dépend de la condition initiale. Pourtant, la fonction f est particulièrement simple !

Résolution en MAPLE. Le logiciel MAPLE fournit un solveur d'équations différentielles sous forme close. Le solveur ne résout pas toutes les équations : la plupart d'entre elles n'ont pas de solution sous forme close, c'est-à-dire de solution qui puisse être notée par une formule finie ne faisant figurer que les fonctions bien connues (polynômes, sinus, exponentielles, etc.). Le solveur obtient toutefois un résultat sur cette équation-ci :

```
> edo := diff (y(x),x) = y(x)^2;
                                  d             2
                          edo := -- y(x) = y(x)
                                  dx



> dsolve ({edo, y(0)=alpha}, y(x));
                                      alpha
                          y(x) = - -----------
                                   x alpha - 1
```

Le logiciel offre également un intégrateur numérique (par défaut, un intégrateur à pas adaptatif construit sur le schéma de Fehlberg [5] d'ordre 4(5), avec un mécanisme de sortie dense).

5. Erwin Fehlberg (1911-1990).

```
> soln := dsolve ({diff(y(x),x) = y(x)^2, y(0)=1}, y(x), numeric);
                      soln := proc(x_rkf45)  ...  end proc
```

La solution retournée par ce solveur est un sous-programme MAPLE capable d'approximer numériquement $y(x)$. Ce sous-programme ne contient pas la solution sous forme close : il est très proche des programmes FORTRAN que nous allons écrire. Numériquement, il rencontre toutefois une difficulté en $x = 1$:

```
> soln (0);
                                [x = 0., y(x) = 1.]

> soln (0.9);
                        [x = 0.9, y(x) = 10.0000034789066]

> soln (1);
Error, (in soln) cannot evaluate the solution further right of .99999999,
probably a singularity
```

Solutions multiples. Considérons maintenant la fonction [5, Example 1.3]

$$f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}\,, \quad (x, y) \mapsto \sqrt{y}\,,$$

qui définit l'équation différentielle

$$\dot{y} = \sqrt{y}\,.$$

Pour une condition initiale $y(0) = \alpha$ strictement positive, la solution est unique :

$$y(x) = \frac{1}{4}\,(x + 2\sqrt{\alpha})^2\,.$$

Toutefois, pour la condition initiale $y(0) = 0$, il existe deux solutions :

$$y(x) = \frac{1}{4}\,x^2 \quad \text{ou bien} \quad y(x) = 0\,.$$

Ces deux solutions ont même vecteur tangent en $x = 0$. L'existence de deux solutions différentes pour une même condition initiale est liée au fait que la dérivée partielle $\partial f/\partial y$, qui vaut $1/\sqrt{y}$, est infinie pour $y = 0$.

Conclusion. Mathématiquement, les questions d'existence et d'unicité des solutions des équations différentielles sont difficiles. Dans ce cours, on les évitera la plupart du temps mais il est important de garder leur existence à l'esprit. C'est souvent à cause d'elles que certains énoncés contiennent des locutions telles que en général ou sous certaines hypothèses .

4.2.1 Systèmes d'équations différentielles

Dans le cas d'un système, la variable y devient un vecteur de $\mathbb{R}^n$ et la fonction f devient un vecteur de n fonctions

$$f_i : [x_0,\, x_{\text{end}}] \times \mathbb{R}^n \to \mathbb{R}\,, \quad 1 \leqslant i \leqslant n\,.$$

Le système de n équations différentielles ainsi défini s'écrit :

$$\dot{y}_1 = f_1(x, y_1, y_2, \ldots, y_n)\,, \quad \dot{y}_2 = f_2(x, y_1, y_2, \ldots, y_n)\,, \quad \ldots\,, \quad \dot{y}_n = f_n(x, y_1, y_2, \ldots, y_n)\,.$$

Sa solution est, en général, une famille de vecteurs de n fonctions $y(x) = (y_1(x), y_2(x), \ldots, y_n(x))^T$. Pour fixer l'une des solutions de cette famille, il suffit, en général, de fixer un vecteur de n conditions initiales $y(x_0) = (\alpha_1, \alpha_2, \ldots, \alpha_n)^T$.

Deux prédateurs se disputant pour la même proie. Les deux fonctions suivantes

$$f_1 : (x, y_1, y_2) \mapsto y_1\,(1 - y_1) - y_1\,y_2\,, \quad f_2 : (x, y_1, y_2) \mapsto 2\,y_2\,\left(1 - \frac{y_2}{2}\right) - 3\,y_1\,y_2$$

définissent le système de deux équations différentielles autonome suivant :

$$\dot{y}_1 = y_1\,(1 - y_1) - y_1\,y_2\,, \quad \dot{y}_2 = 2\,y_2\,\left(1 - \frac{y_2}{2}\right) - 3\,y_1\,y_2\,.$$

C'est un cas particulier de système décrivant l'évolution de la population de deux espèces se disputant pour la même proie (la proie n'apparaît pas dans le modèle). Les variables $y_1(x)$ et $y_2(x)$ donnent les populations des deux espèces à l'instant x. Chaque fonction f_i est construite à partir d'une variante de l'équation logistique (les populations $y_1(x)$ et $y_2(x)$ vont demeurer dans les intervalles $[0,\, 1]$ et $[0,\, 2]$). Les termes en $y_1\,y_2$ modélisent l'effet d'une rencontre entre les représentants des deux espèces, qui tourne au désavantage de l'espèce y_2.

Équation différentielle d'ordre deux. Toute équation différentielle d'ordre deux est équivalente à un système de deux équations différentielles d'ordre 1. Prenons l'exemple de l'équation

$$m\,\ddot{y}_1 + c\,\dot{y}_1 + k\,y_1 = 0\,,$$

qui décrit la dynamique d'une masse m attachée à un ressort de constante de raideur k, avec un coefficient de frottement c. La variable dépendante $y_1(x)$ donne la position de la masse par rapport à sa position d'équilibre. En renommant $y_2 = \dot{y}_1$, on obtient le système suivant, où la première équation code le renommage de la dérivée première de y_1 :

$$\dot{y}_1 = y_2\,, \quad \dot{y}_2 = -\frac{k}{m}\,y_1 - \frac{c}{m}\,y_2\,.$$

Le problème restreint à trois corps. Le système (4.1) est un système de deux équations différentielles d'ordre deux, en deux variables y_1 et y_2. Ce système est équivalent à un système de quatre équations différentielles d'ordre un, en ajoutant deux nouvelles variables $y_3 = \dot{y}_1$ et $y_4 = \dot{y}_2$, comme dans le cas de système masse-ressort. La nouvelle formulation du système (4.1) est donnée ci-dessous :

$$\begin{aligned}
\dot{y}_1 &= y_3\,, \\
\dot{y}_2 &= y_4\,, \\
\dot{y}_3 &= y_1 + 2\,y_4 - \mu' \frac{y_1+\mu}{D_1} - \mu \frac{y_1-\mu'}{D_2}\,, \\
\dot{y}_4 &= y_2 - 2\,y_3 - \mu' \frac{y_2}{D_1} - \mu \frac{y_2}{D_2}\,, \\
D_1 &= ((y_1+\mu)^2 + y_2^2)^{3/2}\,, \\
D_2 &= ((y_1-\mu')^2 + y_2^2)^{3/2}\,.
\end{aligned} \tag{4.3}$$

4.3 Les schémas élémentaires

4.3.1 Le schéma d'Euler

On commence par le cas d'une seule variable. La donnée du problème est une fonction $f : [x_0,\, x_{\text{end}}] \times \mathbb{R} \to \mathbb{R}$, un réel α et un nombre de pas $N \geqslant 1$. Le problème de Cauchy à résoudre s'écrit :

$$\dot{y} = f(x,y)\,, \quad y(x_0) = \alpha\,. \tag{4.4}$$

La méthode d'Euler[6] est une méthode à pas de longueur fixe

$$h = \frac{x_{\text{end}} - x_0}{N}.$$

Elle consiste à calculer une suite de $N+1$ points (x_i, y_i), pour $0 \leqslant i \leqslant N$. Ces points définissent un polygone qui approxime le graphe de la solution exacte $y(x)$. Sous certaines conditions, le polygone d'Euler converge vers le graphe de la solution exacte, dans le sens où, plus le nombre de pas N augmente, plus l'ordonnée y_N du dernier point calculé est proche de l'ordonnée $y(x_N) = y(x_{\text{end}})$ du graphe de la solution exacte, en $x = x_N = x_{\text{end}}$ (attention à ne pas confondre y_i, l'ordonnée du $i$ème point calculé et $y(x_i)$, l'ordonnée du point exact en $x = x_i$).

La fonction f définit un *champ de vecteurs* dans la bande $[x_0, x_{\text{end}}] \times \mathbb{R}$. Les vecteurs de ce champ sont définis en chaque point (x,y) de la bande. Ils sont de coordonnées :

$$\begin{pmatrix} \Delta\, x \\ \Delta\, y \end{pmatrix} = \begin{pmatrix} 1 \\ f(x,y) \end{pmatrix} \tag{4.5}$$

Ce champ de vecteurs donne les tangentes aux graphes de toutes les solutions du problème (4.4) (pour toutes les valeurs possibles de α). Un exemple est donné en Figure 4.3. Voir Figure 4.2 pour un exemple de graphe d'une solution, dans le cas de deux variables.

6. Leonhard Euler (1707-1783).

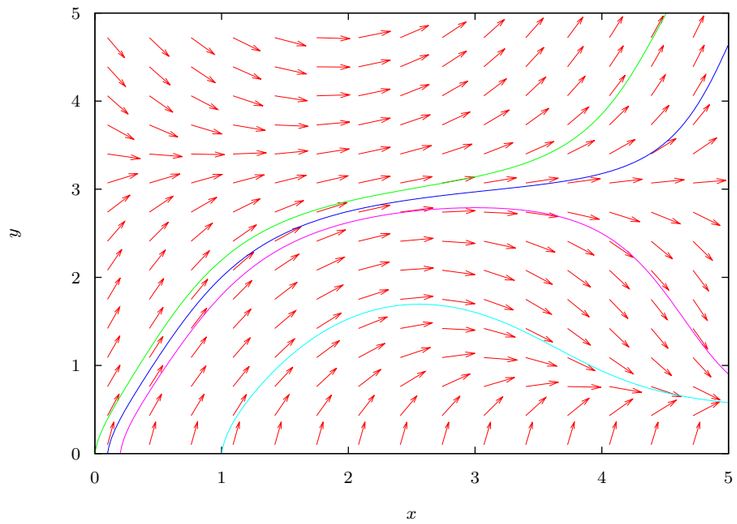

FIGURE 4.3 – Champ de vecteurs (4.5) et graphes de quelques solutions (pour quelques conditions initiales différentes) de l'équation différentielle $\dot{y} = 1/(y + 0.1) - (x - 2)\sin(y)$. Les longueurs des vecteurs du champ ont été normalisées. On observe que les graphes des solutions suivent les vecteurs.

La méthode d'Euler est alors évidente : à partir du point $(x_0, y_0) = (x_0, \alpha)$, on obtient (x_1, y_1) en suivant le vecteur défini en ce point sur une distance h :

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + h \begin{pmatrix} 1 \\ f(x_0, y_0) \end{pmatrix} . \tag{4.6}$$

Il suffit ensuite de remplacer (x_0, y_0) par (x_1, y_1) et d'appliquer à nouveau le formule jusqu'à atteindre $x_N = x_{\text{end}}$. On peut également utiliser la formulation équivalente ci-dessous (les abscisses sont données par une formule très simple ; les ordonnées sont définies par une suite récurrente) :

$$x_i = x_0 + h\, i\,, \quad y_0 = \alpha, \quad y_i = y_{i-1} + h\, f(x_{i-1}, y_{i-1}) \quad (1 \leqslant i \leqslant N)\,.$$

Exemple. Pour illustrer la méthode, on peut prendre le système défini par $f : (x, y) \mapsto y$ et $\alpha = 1$ sur l'intervalle $[x_0, x_{\text{end}}] = [0, 1]$. Le problème de Cauchy est alors : $\dot{y} = y$ avec $y(0) = 1$. La solution est $y(x) = e^x$. En quelque sorte, on cherche à calculer $y(x_{\text{end}}) = e \simeq 2.7182818$. Commençons par appliquer la méthode pour $N = 1$ (un seul pas de longueur $h = 1$). En appliquant la formule (4.6), on trouve l'approximation $e \simeq y_1 = 2$:

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} + h \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} .$$

Recommençons avec $N = 2$ (deux pas de longueur $h = 1/2$). On trouve une meilleure approximation, $e \simeq y_2 = 2.25$. Voir aussi Figure 4.4.

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} + h \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \frac{3}{2} \end{pmatrix}, \quad \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \frac{3}{2} \end{pmatrix} + h \begin{pmatrix} 1 \\ \frac{3}{2} \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{9}{4} \end{pmatrix}.$$

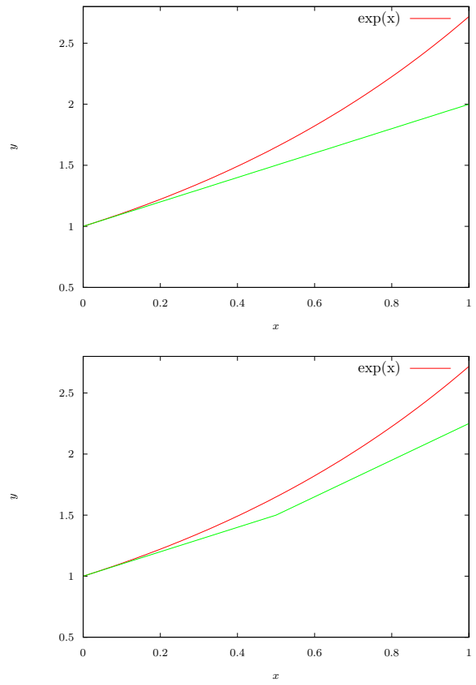

FIGURE 4.4 – Résolution numérique du problème de Cauchy $\dot{y} = y$ avec $y(0) = 1$, par la méthode d'Euler. Intégration avec $N = 1$ pas en haut et $N = 2$ pas en bas.

Le programme FORTRAN suivant poursuit l'exemple pour [7] $N = 2^0, 2^1, \ldots, 2^{10}$.

```
PROGRAM EULER
IMPLICIT NONE
DOUBLE PRECISION Y_XEND, X0, XEND, Y0, H, X, Y, FCN
INTEGER I, K, NBSTEPS
Y_XEND = EXP(1D0)
```

7. Dans les programmes FORTRAN, on note `NBSTEPS` au lieu de `N`, qu'on garde pour désigner la dimension n des systèmes d'équations différentielles.

```
      X0 = 0D0
      XEND = 1D0
      Y0 = 1D0
      WRITE (*,*)
     $  '    NBSTEPS', '            Y',
     $  '               NB BITS EXACTS (APPROX.)'
      DO K = 1,10
         NBSTEPS = INT(2**K)
         H = (XEND - X0)/NBSTEPS
         X = X0
         Y = Y0
         DO I = 1, NBSTEPS
* L'ordre des deux instructions suivantes est important
            Y = Y + H*FCN(X,Y)
            X = X0 + H*I
         END DO
         WRITE (*,*) NBSTEPS, Y, -(LOG(ABS((Y_XEND-Y)/Y_XEND)))/LOG(2D0)
      END DO
      END PROGRAM

      DOUBLE PRECISION FUNCTION FCN (X, Y)
      IMPLICIT NONE
      DOUBLE PRECISION X, Y
      FCN = Y
      END FUNCTION
```

Pour chaque valeur de N, il affiche une approximation du nombre de bits exacts de y_N, c'est-à-dire :

$$\text{nombre de bits exacts} \simeq -\log_2(\text{erreur relative}) = -\log_2\left(\frac{|y(x_{\text{end}}) - y_N|}{|y(x_{\text{end}})|}\right) .$$

Voici le résultat de la simulation

NBSTEPS	Y	NB BITS EXACTS (APPROX.)
2	2.2500000000000000	2.5372460800170895
4	2.4414062500000000	3.2953853279718790
8	2.5657845139503479	4.1558392988653825
16	2.6379284973665995	5.0801933977908478
32	2.6769901293781833	6.0406994465622814
64	2.6973449525650999	7.0205050447733184
128	2.7077390196880193	8.0102919557002483
256	2.7129916242534331	9.0051559130912082
512	2.7156320001689900	10.002580450068470
1024	2.7169557294664357	11.001290849636048

On observe qu'à chaque fois qu'on divise la longueur du pas, h, par deux, on gagne *un* bit exact en $x = x_{\text{end}}$ (l'erreur relative en $x = x_{\text{end}}$ — l'erreur relative *globale* — est divisée par deux). Cette erreur est donc en h^1. La méthode d'Euler est dite d'ordre *un* . Ce résultat peut paraître acceptable sur un exemple aussi simple mais il est insuffisant pour traiter un exemple un peu réaliste, comme le montre la Figure 4.5.

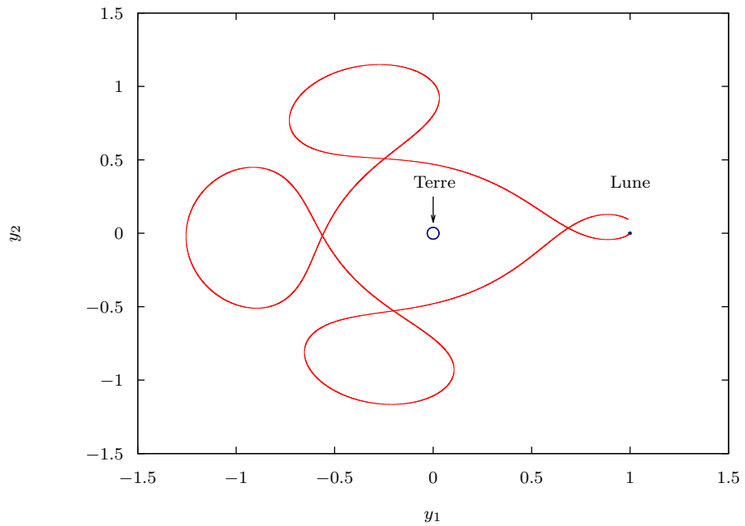

FIGURE 4.5 – La même orbite qu'en Figure 4.1, calculée avec la méthode d'Euler et un million de pas. L'erreur relative en $x = x_{\text{end}}$ vaut 10^{-1}. L'orbite devrait être fermée. La méthode d'Euler ne permet pas de s'en rendre compte, malgré le nombre de pas utilisés.

Question 1. Appliquer la méthode d'Euler au problème de Cauchy : $\dot{y} = y - 2\,x/y$ avec $y(x_0) = 1$ sur l'intervalle $[x_0,\, x_{\text{end}}] = [0,\, 2]$ pour $N = 1$, puis $N = 2$ pas. La solution exacte est $y(x) = \sqrt{2\,x + 1}$.

4.3.2 Le schéma de Runge

Dans cette section, on continue de présenter informellement la notion d'ordre avant de donner des définitions plus précises dans la section suivante. Le schéma de Runge[8] est appelé aussi schéma du point du milieu . L'idée est la suivante : si on applique le schéma d'Euler sur un point (x_0, y_0), on aboutit à un point $(x_0 + h, y_0 + h\,k_1)$, avec $k_1 = f(x_0, y_0)$. Ces deux points définissent un rectangle (dans le cas d'équations différentielles en une variable, en tous cas). Le milieu de ce rectangle est le point de coordonnées :

$$\left(x_0 + \frac{h}{2}\,,\ y_0 + h\,\frac{k_1}{2}\right)\,.$$

En ce point se trouve un vecteur du champ de vecteurs, qui a pour coordonnées :

$$\begin{pmatrix} \Delta\,x \\ \Delta\,y \end{pmatrix} = \begin{pmatrix} 1 \\ k_2 \end{pmatrix}\,, \quad k_2 = f\left(x_0 + \frac{h}{2}\,,\ y_0 + h\,\frac{k_1}{2}\right)\,.$$

8. Carl David Tolmé Runge (1856-1927).

La méthode de Runge consiste à suivre ce vecteur, depuis (x_0, y_0), suivant un pas de longueur h (voir Figure 4.6) :

$$y_1 = y_0 + h\, k_2\,.$$

Il est facile de modifier le programme précédent pour qu'il utilise le schéma de Runge plutôt que

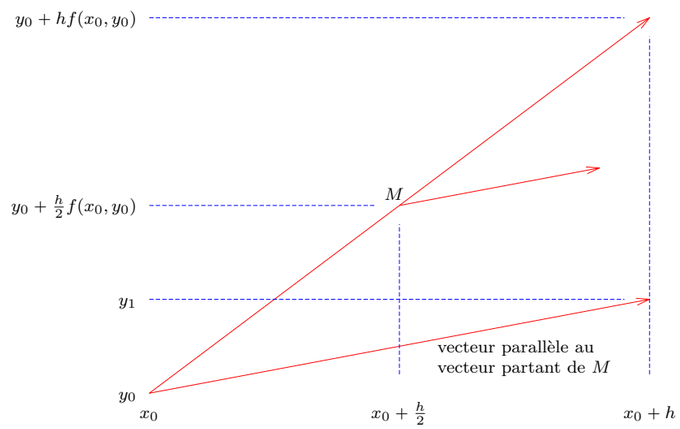

FIGURE 4.6 – Interprétation graphique du schéma de Runge. Un pas avec la méthode d'Euler définit un rectangle de milieu M. Le schéma consiste à suivre le vecteur défini en M depuis (x_0, y_0).

celui d'Euler. À l'exécution, on observe qu'à chaque fois qu'on divise la longueur du pas h par deux, l'erreur relative globale est divisée par 4. Cette erreur est donc en h^2. Le schéma de Runge est dit d'ordre *deux* :

```
NBSTEPS          Y                NB BITS EXACTS (APPROX.)
      2   2.6406250000000000        5.1294384425499331
      4   2.6948556900024414        6.8584320694052145
      8   2.7118412385519850        8.7212864917120907
     16   2.7165935224747666        10.652902927432585
     32   2.7178496739802589        12.618860307260315
     64   2.7181725115638296        14.601891029858672
    128   2.7182543383212754        16.593421378239558
    256   2.7182749357407454        18.589190555576408
    512   2.7182801027521668        20.587076175334616
   1024   2.7182813967161392        22.586019232613165
```

Le schéma de Runge est-il plus efficace que celui d'Euler ? Pour que la comparaison soit honnête, il faut comparer les deux schémas à quantité de travail égale. Dans le domaine de l'intégration numérique des équations différentielles, on mesure cette quantité de travail en comptant le nombre d'appels à la fonction f.

On raisonne sur l'exemple. Pour le schéma d'Euler, on obtient b bits de précisions en $x = x_{\text{end}}$ avec approximativement 2^{b-1} pas, c'est-à-dire 2^{b-1} évaluations, puisque chaque pas coûte une évaluation. Pour le schéma de Runge, on obtient b bits de précisions avec approximativement $2^{b/2-1}$ pas, c'est-à-dire $2 \times 2^{b/2-1} = 2^{b/2}$ évaluations, puisque chaque pas coûte deux évaluations. À précision égale, une méthode d'ordre deux effectue donc beaucoup moins de travail qu'une méthode d'ordre un.

4.4 Les schémas de Runge-Kutta

Soit s un entier (le nombre d'étages). La formule suivante, qui calcule y_1 à partir de (x_0, y_0) est appelée *schéma de Runge-Kutta explicite*[9] *à s étages* [2, Definition 1.1, page 134]. La lettre h désigne le pas. Les coefficients c_i, b_j et a_{ij} sont des réels :

$$\begin{aligned}
k_1 &= f(x_0,\, y_0)\,, \\
k_2 &= f(x_0 + c_2\, h,\, y_0 + h\, a_{21}\, k_1)\,, \\
k_3 &= f(x_0 + c_3\, h,\, y_0 + h\,(a_{31}\, k_1 + a_{32}\, k_2))\,, \\
&\vdots \\
k_s &= f(x_0 + c_s\, h,\, y_0 + h\,(a_{s1}\, k_1 + a_{s2}\, k_2 + \cdots + a_{s(s-1)}\, k_{s-1}))\,, \\
y_1 &= y_0 + h\,(b_1\, k_1 + b_2\, k_2 + \cdots + b_s\, k_s)\,.
\end{aligned} \tag{4.7}$$

Il est classique de représenter les coefficients d'un schéma de Runge-Kutta[10] par un tableau (le coefficient $c_1 = 0$). Les schémas explicites sont caractérisés par le fait que la matrice des a_{ij} est triangulaire inférieure :

$$\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{21} & & & & \\
c_3 & a_{31} & a_{32} & & & \\
\vdots & \vdots & & \ddots & & \\
c_s & a_{s1} & a_{s2} & \cdots & a_{s(s-1)} & \\
\hline
 & b_1 & b_2 & \cdots & b_{s-1} & b_s
\end{array}$$

Les méthodes historiques d'Euler ($s = 1$), de Runge ($s = 2$), de Heun[11] ($s = 3$), ainsi que le schéma RK4 ($s = 4$) sont donnés par les tableaux suivants. Tous ces schémas sont d'ordre

9. Le schéma est *explicite* parce que chaque k_i peut être calculé à partir de $k_1, k_2, \ldots, k_{i-1}$.
10. Martin Wilhelm Kutta (1867-1944).
11. Karl Heun (1859-1929).

$p = s$ mais on rencontrera d'autres schémas d'ordre $p < s$:

$$\begin{array}{c|c} 0 & \\ \hline & 1 \end{array} \qquad \begin{array}{c|cc} 0 & & \\ \frac{1}{2} & \frac{1}{2} & \\ \hline & 0 & 1 \end{array} \qquad \begin{array}{c|ccc} 0 & & & \\ \frac{1}{3} & \frac{1}{3} & & \\ \frac{2}{3} & 0 & \frac{2}{3} & \\ \hline & \frac{1}{4} & 0 & \frac{3}{4} \end{array} \qquad \begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

Les coefficients ne sont pas quelconques : ils sont choisis de façon à fixer l'ordre du schéma.

4.4.1 Erreur locale et erreur globale

Avant de passer à l'étude plus rigoureuse de la notion d'ordre, il est utile de bien distinguer l'erreur globale de l'erreur locale. L'erreur la plus facile à appréhender, celle qui se voit le mieux lorsqu'on traite des exemples par programme, c'est l'erreur globale, c'est-à-dire l'erreur en $x = x_{\text{end}}$. Le point exact (son ordonnée, en fait) est $y(x_{\text{end}})$. En supposant qu'on a effectué N pas d'intégration, le dernier point calculé est y_N. On a donc

$$\text{erreur globale} = |y(x_{\text{end}}) - y_N| \,.$$

L'erreur locale est l'erreur après seulement un pas d'intégration. Elle est moins facile à visualiser expérimentalement mais nettement plus pratique à manipuler lorsqu'on analyse les schémas rigoureusement. C'est donc elle qui est naturellement utilisée pour définir l'ordre d'un schéma. Pour l'introduire, on part du point courant [12] (x_0, y_0). Le point exact du graphe de la fonction $y(x)$ qui passe par le point courant est $y(x_1) = y(x_0 + h)$. Le point calculé est y_1. On a donc :

$$\text{erreur locale} = |y(x_1) - y_1| \,.$$

Dans les deux cas, il est parfois utile de distinguer l'erreur relative de l'erreur absolue.

4.4.2 Ordre d'un schéma de Runge-Kutta

Définition 6 [2, Definition 1.2, page 134] *Un schéma de Runge-Kutta* (4.7) *est dit* d'ordre p *si (en supposant la fonction f suffisamment dérivable), il existe une constante $K > 0$ telle que l'erreur locale vérifie :*

$$|y(x_0 + h) - y_1| \quad \leqslant \quad K\, h^{p+1} \,. \tag{4.8}$$

Lors de la présentation informelle de la notion d'ordre, c'est l'erreur globale que nous avons observée. Comment relier ces observations à la définition ?

12. La notation (x_0, y_0) pour le point courant a le mérite d'alléger les notations mais constitue un léger glissement de sens par rapport aux notations introduites dans les sections précédentes. On pourrait le justifier en posant qu'on change l'intervalle d'intégration considéré.

Schéma d'Euler. Nous avons observé que l'erreur globale semble être en h (quand on divise h par deux, l'erreur globale est approximativement divisée par deux). Nous avons affirmé (sans preuve) que le schéma d'Euler est d'ordre $p = 1$. Or, d'après la définition, l'erreur locale est en h^2. Il semble donc qu'une erreur locale en h^2 corresponde à une erreur globale en h.

Schéma de Runge. Nous avons observé que l'erreur globale semble être en h^2. (quand on divise h par deux, l'erreur globale est approximativement divisée par deux au carré). Nous avons affirmé (sans preuve) que le schéma de Runge est d'ordre $p = 2$. Or, d'après la définition, l'erreur locale est en h^3. Il semble donc à nouveau que l'exposant chute de un et qu'une erreur locale en h^3 corresponde à une erreur globale en h^2.

Généralisation. Ces observations se généralisent (voir [2, Theorem 3.4, page 160 ; Theorem 3.6, page 162] pour des énoncés plus précis et plus généraux) :

Proposition 8 *Si on intègre* (4.4) *de* x_0 *à* x_{end} *avec un pas* h*, au moyen d'un schéma de Runge-Kutta* (4.7) *d'ordre* p *alors, sous certaines hypothèses, il existe une constante* $K > 0$ *(qui dépend de* f *et de la différence* $x_{\text{end}} - x_0$*) telle que l'erreur globale vérifie :*

$$|y(x_{\text{end}}) - y_N| \quad \leqslant \quad K\, h^p\,.$$

4.4.3 Détermination de l'ordre

Pour prouver qu'un schéma de Runge-Kutta est d'ordre p, l'idée consiste à comparer deux développements en série et vérifier qu'ils ont mêmes coefficients jusqu'au terme en h^p inclus. Cette condition se traduit par des équations que les coefficients b_i, c_j et a_{ij} doivent satisfaire pour fournir un schéma d'ordre p.

Schéma d'Euler. Le premier développement en série est le développement de Taylor, en $x = x_0$ de la solution exacte $y(x)$. La formule générale du développement de Taylor d'une fonction $y(x)$ est (la dérivation se fait par rapport à la variable, x) :

$$y(x_0 + h) \quad = \quad y(x_0) + h\,\dot{y}(x_0) + \frac{1}{2}\,h^2\,\ddot{y}(x_0) + \frac{1}{6}\,h^3\,\dddot{y}(x_0) + \cdots \tag{4.9}$$

Dans le cas qui nous intéresse, la fonction $y(x)$ est solution de l'équation différentielle $\dot{y}(x) = f(x, y(x))$. Sans avoir besoin de résoudre en aucune façon cette équation, il est possible de remplacer $\dot{y}(x_0)$ par $f(x_0, y(x_0))$ dans (4.9). On obtient :

$$y(x_0 + h) \quad = \quad y(x_0) + h\,f(x_0, y(x_0)) + \frac{1}{2}\,h^2\,\frac{\mathrm{d}f}{\mathrm{d}x}\,(x_0, y(x_0)) + \cdots \tag{4.10}$$

La condition initiale est $y(x_0) = y_0$. En reportant cette hypothèse dans la formule, on obtient, pour la solution exacte :

$$y(x_0 + h) \quad = \quad y_0 + h\,f(x_0, y_0) + \frac{1}{2}\,h^2\,\frac{\mathrm{d}f}{\mathrm{d}x}\,(x_0, y_0) + \cdots$$

Le deuxième développement en série est donné par le schéma d'Euler à partir du point (x_0, y_0) :

$$y_1 \quad = \quad y_0 + h\, f(x_0, y_0)\,. \tag{4.11}$$

On voit que la différence entre ces deux développements (l'erreur locale) est en h^2. Sous réserve que h ne soit pas trop grand (il ne doit pas être plus grand que 1, par exemple) et que les valeurs des dérivées de f soient bornées, on conclut l'existence de la constante K de la définition 6.

Le schéma de Runge. C'est la même idée. On commence les calculs à la main pour voir les mécanismes mis en œuvre. On conclut avec un logiciel de calcul formel parce que c'est trop pénible à faire à la main [13]. Pour prolonger le développement (4.10) de la solution exacte, il suffit d'appliquer la formule pour la dérivée des fonctions composées. La fonction f dépend de deux variables qu'on a notées x et y. Ses dérivées partielles sont donc

$$f_x = \frac{\partial f}{\partial x}\,(x, y) \quad \text{et} \quad f_y = \frac{\partial f}{\partial y}\,(x, y)\,.$$

Si $u(x)$ et $v(x)$ sont deux fonctions dérivables, on a donc (dérivation des fonctions composées de plusieurs variables) :

$$\frac{\mathrm{d}f}{\mathrm{d}x}\,(u(x), v(x)) \quad = \quad f_x(u(x), v(x))\,\dot{u}(x) + f_y(u(x), v(x))\,\dot{v}(x)\,.$$

Dans le cas qui nous intéresse, $u(x) = x$ et $v(x) = y(x)$. Par conséquent, $\dot{u}(x) = 1$ et $\dot{v}(x) = f(x, y(x))$ et on a :

$$\begin{aligned} \frac{\mathrm{d}f}{\mathrm{d}x}\,(x, y(x)) \quad &= \quad f_x\,(x, y(x)) + f_y\,(x, y(x))\, f(x, y(x))\,, \\ &= \quad (f_x + f_y\, f)(x, y(x))\,. \end{aligned}$$

On obtient donc, pour la solution exacte :

$$y(x_0 + h) \quad = \quad y_0 + h\, f(x_0, y_0) + \frac{1}{2}\, h^2\, (f_x + f_y\, f)(x_0, y_0) + \cdots \tag{4.12}$$

Le deuxième développement en série est donné par le schéma de Runge à partir du point (x_0, y_0) :

$$y_1 \quad = \quad y_0 + h\, f\left(x_0 + \frac{1}{2}\, h,\; y_0 + \frac{1}{2}\, h\, f(x_0, y_0)\right)\,. \tag{4.13}$$

Pour pouvoir comparer les deux développements, il suffit de réécrire cette formule, qui dépend déjà du pas h, sous la forme d'une série en la variable h. On voit donc y_1 comme une fonction de h et on cherche les coefficients de la formule suivante (les dérivations se font par rapport à h) :

$$y_1(h) \quad = \quad y_1(0) + h\, \dot{y}_1(0) + \frac{1}{2}\, h^2\, \ddot{y}_1(0) + \cdots \tag{4.14}$$

13. L'introduction des calculatrices à l'école primaire a changé la façon d'enseigner les mathématiques : il est toujours utile de savoir faire ses quatre opérations mais il est n'est plus aussi important de savoir effectuer de longues opérations, à la main, sans se tromper. L'introduction du calcul formel dans les enseignements de calcul numérique a (devrait avoir !) le même effet.

Avant de mener les calculs, on introduit quelques notations pour alléger les formules. Notons

$$u = u(h) = x_0 + \frac{1}{2}\,h\,, \quad v = v(h) = y_0 + \frac{1}{2}\,h\,f(x_0, y_0)\,.$$

On a donc :

$$u(0) = x_0\,, \quad v(0) = y_0\,, \quad \dot{u} = \dot{u}(h) = \frac{1}{2}\,, \quad \dot{v} = \dot{v}(h) = \frac{1}{2}\,f(x_0, y_0)\,.$$

La formule (4.13) se réécrit :

$$y_1(h) \;=\; y_0 + h\,f(u, v)\,. \tag{4.15}$$

En évaluant le schéma (4.15) en $h = 0$, on trouve $y_1(0) = y_0$. Pour le terme suivant, on commence par dériver (4.15) par rapport à h :

$$\dot{y}_1(h) \;=\; f(u, v) + h\,\frac{\mathrm{d}f}{\mathrm{d}h}\,(u, v)\,. \tag{4.16}$$

En évaluant cette formule en $h = 0$, on obtient le deuxième terme du développement (4.14) : $\dot{y}_1(0) = f(x_0, y_0)$. Pour obtenir le terme suivant, il suffit de dériver (4.16) par rapport à h.

$$\begin{aligned}
\ddot{y}_1(h) \;&=\; \frac{\mathrm{d}}{\mathrm{d}h}\left[f(u, v) + h\,\frac{\mathrm{d}f}{\mathrm{d}h}\,(u, v)\right] \\
&=\; \frac{\mathrm{d}}{\mathrm{d}h}\left[f(u, v) + h\,(f_x(u, v)\,\dot{u} + f_y(u, v)\,\dot{v})\right] \\
&=\; 2\,(f_x(u, v)\,\dot{u} + f_y(u, v)\,\dot{v}) + h\,\frac{\mathrm{d}}{\mathrm{d}h}(\cdots)\,.
\end{aligned}$$

En évaluant cette formule en $h = 0$, on obtient $\ddot{y}_1(0) = (f_x + f_y\,f)(x_0, y_0)$. On obtient ainsi le développement en série de y_1 jusqu'au terme en h^2 :

$$y_1(h) \;=\; y_0 + h\,f(x_0, y_0) + \frac{1}{2}\,h^2\,(f_x + f_y\,f)(x_0, y_0) + \cdots \tag{4.17}$$

On peut donc vérifier que le développement en série de la solution exacte (4.12) et celui du schéma de Runge (4.17) coïncident [14] jusqu'au terme en h^2 inclus. En fait, les calculs ne sont pas finis : pour montrer que le schéma de Runge est d'ordre deux (et pas trois, quatre ou plus) il reste à montrer que les termes en h^3 des deux développements sont différents. Sur le principe, les calculs sont les mêmes. On les mène en MAPLE.

14. On peut être gêné par le sentiment qu'on compare deux objets différents : le développement de Taylor (4.9) en $x = x_0$ de la fonction $y(x)$ et le développement en série (4.14) en $h = 0$ de la fonction $y_1(h)$. En fait, le développement (4.9) est un cas particulier du développement (4.14). Il suffit de poser $x = x_0 + h$ (la variable x devient donc une fonction de h) et de calculer le développement en série en $h = 0$ de $y(x)$. Lors des calculs, il suffit d'utiliser la formule pour la dérivée des fonctions composées :

$$\frac{\mathrm{d}y}{\mathrm{d}h}(x) = \frac{\mathrm{d}y}{\mathrm{d}x}(x)\,\frac{\mathrm{d}x}{\mathrm{d}h} = \frac{\mathrm{d}y}{\mathrm{d}x}(x)\,.$$

On retrouve bien la série (4.9). On compare donc bien deux objets de même nature : deux développements en série, de deux fonctions de h, en $h = 0$.

Mêmes calculs, en MAPLE. On commence par définir l'équation différentielle.

```
> edo := diff (y(x),x) = f(x,y(x));
                                d
                        edo := -- y(x) = f(x, y(x))
                                dx
```

On définit ensuite s et p.

```
> s := 2:
> p := 2:
```

Bien qu'elles aient l'air compliqué, les commandes suivantes ne font que calculer le développement en série générique de la solution exacte, jusqu'à l'ordre $p+1$. L'expression `D[1](f)` désigne la dérivée partielle de f par rapport à sa première variable ($\partial f/\partial x$ dans notre cas). De même `D[1,2](f)` désigne la dérivée partielle seconde $\partial^2 f/(\partial x\,\partial y)$. Le résultat est affecté à la variable *vsol*.

```
> vsol := dsolve ({edo, y(x0)=y0}, y(x), series, order=p+2):
> vsol := collect (subs (x=x0+h, convert (rhs (vsol), polynom)), h);

vsol := y0 + f(x0, y0) h
 + (1/2 D[1](f)(x0, y0) + 1/2 D[2](f)(x0, y0) f(x0, y0)) h  + (

                                                          2
1/6 D[2](f)(x0, y0) D[1](f)(x0, y0) + 1/6 D[2](f)(x0, y0)  f(x0, y0)

 + 1/6 D[1, 1](f)(x0, y0) + 1/3 D[1, 2](f)(x0, y0) f(x0, y0)

                                    2   3
 + 1/6 D[2, 2](f)(x0, y0) f(x0, y0) ) h
```

On calcule maintenant le développement en série de y_1. On commence par calculer une expression symbolique de y_1 en utilisant le schéma (4.7). Les coefficients b_i, c_j et a_{ij} sont des paramètres, qui apparaissent comme coefficients dans la formule.

```
> k[1] := f(x0,y0):
> for i from 2 to s do
>     k[i] := f(x0 + c[i]*h, y0 + h*add (a[i,j]*k[j], j = 1..i-1))
> end do:
> y1 := y0 + h*add(b[i]*k[i], i = 1..s);

 y1 := y0 + h (b[1] f(x0, y0) + b[2] f(x0 + c[2] h, y0 + h a[2, 1] f(x0, y0)))
```

On demande un développement limité de y_1 en h, jusqu'à l'ordre $p+1$. Le résultat est affecté à la variable *rksol*.

```
> rksol := convert (taylor (y1,h,p+2), polynom);

rksol := y0 + (b[1] f(x0, y0) + b[2] f(x0, y0)) h

                                                                          2
     + b[2] (D[1](f)(x0, y0) c[2] + D[2](f)(x0, y0) a[2, 1] f(x0, y0)) h  +

                                      2
    b[2] (1/2 D[1, 1](f)(x0, y0) c[2]

     + c[2] D[1, 2](f)(x0, y0) a[2, 1] f(x0, y0)

                  2          2                       3
     + 1/2 a[2, 1]  f(x0, y0)  D[2, 2](f)(x0, y0)) h
```

On calcule la différence entre les deux développements en série et on remplace les coefficients par leur valeur, dans le schéma de Runge. On voit que le premier terme non nul de la différence est en h^3 :

```
> difference := rksol - vsol:
> collect (subs (b[1] = 0, c[2]=1/2, a[2,1]=1/2, b[2] = 1, difference), h);

                                    2
(-1/24 D[2, 2](f)(x0, y0) f(x0, y0)  - 1/12 D[1, 2](f)(x0, y0) f(x0, y0)
                        2
     - 1/6 D[2](f)(x0, y0)  f(x0, y0) - 1/24 D[1, 1](f)(x0, y0)

                                          3
     - 1/6 D[2](f)(x0, y0) D[1](f)(x0, y0)) h
```

les commandes suivantes donnent les conditions que les coefficients de Runge-Kutta doivent satisfaire pour obtenir une méthode d'ordre 2. Voici le coefficient de h dans la différence entre les deux développements. Ce coefficient doit être nul quelle que soit f. Il faut donc que $b_1 + b_2 = 1$:

```
> X := indets (difference, function):
> collect (eval (diff (difference, h), h = 0), X);

                          (b[1] + b[2] - 1) f(x0, y0)
```

Voici le coefficient de h^2. Il doit être nul quelle que soit f. Donc $b_2\, a_{21} = b_2\, c_2 = 1/2$:

```
> collect (eval (diff (difference, h, h), h = 0), X);

2 (b[2] a[2, 1] - 1/2) D[2](f)(x0, y0) f(x0, y0)

     + 2 (b[2] c[2] - 1/2) D[1](f)(x0, y0)
```

Question 2. Reproduire le calcul ci-dessus en MAPLE pour vérifier que le schéma de Heun est d'ordre 3 et que le schéma RK4 est d'ordre 4 (il s'agit du plus populaire des schémas de Runge-Kutta d'ordre $p = 4$ mais il en existe d'autres — d'où les guillemets).

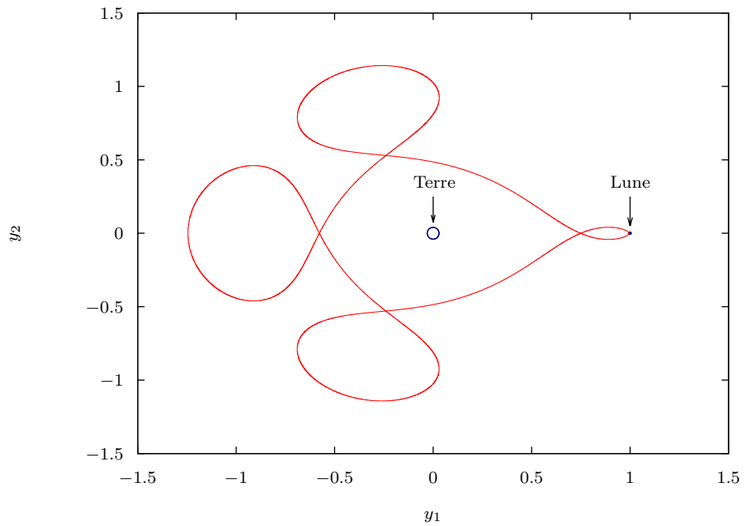

FIGURE 4.7 – La même orbite qu'en Figure 4.1, calculée avec le schéma RK4 et 25000 pas. L'erreur relative en $x = x_{\text{end}}$ vaut 10^{-3}. Comme le schéma comporte $s = 4$ étages, le travail effectué par l'intégrateur est de $4 \times 25000 = 10^5$ appels à la fonction f. Le résultat est bon mais reste coûteux.

Question 3. On souhaite intégrer un système d'équations différentielles. Dans le schéma (4.7), quelles sont les quantités qui restent des scalaires et celles qui deviennent des vecteurs ?

4.5 Intégrateur à pas adaptatif

On voudrait diminuer le travail effectué par l'intégrateur. L'idée consiste à diminuer la longueur des pas là où la courbe effectue des virages serrés et à l'augmenter dans les virages plus larges. Pour cela, il suffit de pouvoir estimer l'erreur locale à chaque pas et d'ajuster le pas en fonction de cette estimation.

Plusieurs techniques sont possibles. On présente celle des formules de Runge-Kutta emboîtées . L'idée consiste à calculer, en même temps que y_1, une autre approximation $\hat{y}_1$, d'ordre $\hat{p} \neq p$. La différence $y_1 - \hat{y}_1$ fournit une estimation de l'erreur locale pour l'approximation la moins précise. Pour éviter d'augmenter le travail de l'intégrateur, l'astuce consiste à calculer $\hat{y}_1$ en utilisant les évaluations de f qui servent à calculer y_1 : les k_i restent donc inchangés et il suffit de trouver un jeu de coefficients $\hat{b}_i$ pour $\hat{y}_1$. Pour les schémas avec une formule emboîtée, le tableau ressemble

donc à :

0					
c_2	a_{21}				
c_3	a_{31}	a_{32}			
$\vdots$	$\vdots$		$\ddots$		
c_s	a_{s1}	a_{s2}	$\cdots$	$a_{s(s-1)}$	
	b_1	b_2	$\cdots$	b_{s-1}	b_s
	$\hat{b}_1$	$\hat{b}_2$	$\cdots$	$\hat{b}_{s-1}$	$\hat{b}_s$

avec

$$y_1 = y_0 + h\left(b_1\, k_1 + b_2\, k_2 + \cdots + b_s\, k_s\right), \quad \hat{y}_1 = y_0 + h\left(\hat{b}_1\, k_1 + \hat{b}_2\, k_2 + \cdots + \hat{b}_s\, k_s\right).$$

Des exemples sont donnés Figure 4.8.

0					
$\frac{1}{2}$	$\frac{1}{2}$				
$\frac{1}{2}$	0	$\frac{1}{2}$			
1	0	0	1		
$\frac{3}{4}$	$\frac{5}{32}$	$\frac{7}{32}$	$\frac{13}{32}$	$-\frac{1}{32}$	
y_1	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$	0
$\hat{y}_1$	$-\frac{1}{2}$	$\frac{7}{3}$	$\frac{7}{3}$	$\frac{13}{6}$	$-\frac{16}{3}$

0						
$\frac{1}{5}$	$\frac{1}{5}$					
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$				
$\frac{4}{5}$	$\frac{44}{45}$	$-\frac{56}{15}$	$\frac{32}{9}$			
$\frac{8}{9}$	$\frac{19372}{6561}$	$-\frac{25360}{2187}$	$\frac{64448}{6561}$	$-\frac{212}{729}$		
1	$\frac{9017}{3168}$	$-\frac{355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$-\frac{5103}{18656}$	
1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$
y_1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$
$\hat{y}_1$	$\frac{5179}{57600}$	0	$\frac{7571}{16695}$	$\frac{393}{640}$	$-\frac{92097}{339200}$	$\frac{187}{2100}$

FIGURE 4.8 – Schémas de Runge-Kutta avec formules emboîtées. À gauche le schéma Zonneveld 4(3) (la notation $p(\hat{p})$ donne l'ordre de la formule principale et, entre parenthèses, celui de la formule emboîtée) à $s = 5$ étages. À droite, le schéma Dormand-Prince 5(4) à $s = 7$ étages (celui qui a été utilisé pour la Figure 4.1). Voir [2, pages 167 et 178]

4.5.1 Contrôle du pas en pratique

L'intégrateur n'est plus paramétré par un nombre de pas N mais par une précision souhaitée $\varepsilon > 0$ pour l'erreur locale (aussi bien relative qu'absolue) [15].

15. Dans les intégrateurs écrits avec plus de soin, on peut avoir une précision souhaitée par composante du vecteur y et prévoir deux $\varepsilon > 0$ par composante : un pour l'erreur relative et l'autre pour l'erreur absolue. Voir [2, page 167].

Formule pour le pas — l'idée. Voir [4, chapitre III, page 66] ou [2, page 167]. Dès qu'un pas h a été choisi, l'intégrateur calcule y_1 et $\hat{y}_1$ à partir de y_0. Une estimation de l'erreur pour le résultat le moins précis [16] est (on compliquera la formule en (4.20) un peu plus bas) :

$$\textit{erreur} \underset{\text{def}}{=} |y_1 - \hat{y}_1|. \tag{4.18}$$

Posons $q = \min(p, \hat{p})$. En tant que fonctions du pas h, les développements en série de y_1 et $\hat{y}_1$ coïncident jusqu'au terme en h^q inclus. On peut donc partir du principe que :

$$\textit{erreur} \simeq c\, h^{q+1}.$$

On cherche une nouvelle valeur h_{new} pour le pas telle que

$$\varepsilon = c\, h_{\text{new}}^{q+1}.$$

On trouve (on compliquera cette formule en (4.21) plus bas) :

$$h_{\text{new}} = h \left(\frac{\varepsilon}{\textit{erreur}}\right)^{\frac{1}{q+1}}. \tag{4.19}$$

On remarque que, si $\textit{erreur} < \varepsilon$ alors $h_{\text{new}} > h$: on peut accepter le pas h et augmenter sa longueur pour l'itération suivante. Inversement, si $\textit{erreur} > \varepsilon$ alors $h_{\text{new}} < h$: on doit refuser le pas et diminuer sa longueur. Un pseudo-code pour l'intégrateur est donné Figure 4.9.

```
procedure intégrateur à pas adaptatif (x_0, x_end, ε)
  h := min(10^-3, x_end − x_0)
  while x_0 < x_end do
    Calculer y_1 et ŷ_1
    Calculer erreur (formule 4.20) et h_new (formule 4.21)
    if erreur ⩽ ε then
      x_0 := x_0 + h (le pas est accepté)
      y_0 := y_1
      h := min(h_new, x_end − x_0)
    else
      h := h_new (le pas est rejeté)
    end if
  end do
end
```

FIGURE 4.9 – Schéma d'intégrateur à pas adaptatif. Voir [4, chapitre III, page 67]. On remarque que les pas acceptés et rejetés ne sont pas gérés de façon symétrique.

16. Pour les deux schémas que nous avons donné, y_1 est plus précis que $\hat{y}_1$ mais ce n'est pas toujours le cas. Un célèbre schéma, dû à Fehlberg, est d'ordre 4(5).

Formule pour le pas — la pratique. La formule (4.18) est un peu trop simple. En pratique, on considère une erreur relative plutôt qu'absolue et on s'arrange pour éviter les divisions par des nombres trop proches de zéro. Dans le cas d'un système de dimension n, on remplace la valeur absolue par une norme :

$$erreur \underset{\text{def}}{=} \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_{1i}-\hat{y}_{1i}}{1+\max(|y_{0i}|,|y_{1i}|)}\right)^2}. \tag{4.20}$$

On essaie aussi d'éviter que h_{new} soit trop différent de h et on préfère prendre une valeur un peu trop petite que trop grande de façon à réduire le risque que le pas soit rejeté à la prochaine itération. En prenant par exemple $facmin = 1/5$, $facmax = 5$ et $fac = 0.9$, on peut utiliser la formule suivante au lieu de (4.19) :

$$h_{\text{new}} = h \min\left(facmax, \max\left(facmin, fac\left(\frac{\varepsilon}{erreur}\right)^{\frac{1}{q+1}}\right)\right). \tag{4.21}$$

Pas initial. Quelle valeur initiale donner au pas ? Des méthodes empiriques ont été suggerées [2, page 169] mais ne sont pas très satisfaisantes. Dans notre cas, il suffit d'initialiser le pas à une petite valeur (10^{-3}). Le mécanisme de l'intégrateur ajustera cette valeur en quelques itérations.

L'exemple. La courbe polygonale de la Figure 4.1 a été calculée avec l'algorithme décrit ci-dessus et le schéma Dormand-Prince 5(4). La précision demandée était $\varepsilon = 1.5\ 10^{-4}$. Il y a eu 54 pas acceptés et 20 pas rejetés et donc un travail total de $(54 + 20) \times s = (54 + 20) \times 7 = 518$ évaluations de la fonction f.

4.5.2 Sortie dense

Les intégrateurs à pas adaptatif sont susceptibles d'effectuer de grands pas, donnant à l'affichage des courbes pas très lisses. L'idée de la sortie dense consiste à calculer, après chaque pas de l'intégrateur, un polynôme $y_\theta(\theta)$ qui approxime $y(x_0 + \theta\, h)$ pour $0 \leqslant \theta \leqslant 1$. Pour lisser les courbes, il suffira, entre deux pas de l'intégrateur, d'évaluer $y_\theta(\theta)$ pour suffisamment de valeurs de θ prises entre 0 et 1. Au delà de cet intérêt graphique, la question de la sortie dense est très importante pour le traitement de discontinuités ou l'intégration d'équations différentielles avec retards [17]. Voir [2, chapters II.6, page 188 et II.17, page 342]. Plusieurs approches sont possibles pour mettre au point un tel polynôme $y_\theta(\theta)$ [2, pages 188-194] et [1, pages 98-115]. On décrit une solution simple mais pas optimale.

La formule. À partir du moment où le schéma considéré est d'ordre $p \geqslant 3$, il existe un polynôme $y_\theta(\theta)$ générique qui fournit une approximation d'ordre 3 de la solution exacte, globalement C^1,

17. Un exemple d'équation intégrale avec retard est donné par (3), page 2.

c'est-à-dire lisse, même aux points de raccord des morceaux de courbe. Ce polynôme vaut [2, page 190] :

$$\begin{aligned} y_\theta &= (1-\theta)\,y_0 + \theta\,y_1 + \\ &\quad \theta\,(\theta-1)\,\big((1-2\,\theta)\,(y_1-y_0) + (\theta-1)\,h\,f(x_0,y_0) + \theta\,h\,f(x_1,y_1)\big) \end{aligned} \tag{4.22}$$

La valeur $f(x_0, y_0)$ est déjà disponible (c'est k_1). Il est uniquement nécessaire de calculer $f(x_1, y_1)$, si cette valeur n'est pas déjà disponible. En effet, plusieurs schémas utilisent cette valeur (quand c'est le cas, elle est souvent dans k_s). De plus, si le pas est accepté, cette valeur peut être réutilisée à l'étape suivante comme valeur de k_1. Lisser la courbe augmente donc très peu le travail de l'intégrateur. Un exemple de courbe lissée apparaît Figure 4.1.

La courbe est globalement lisse. On remarque que $y_\theta(0) = y_0$ et $y_\theta(1) = y_1$. La courbe est donc globalement C^0, c'est-à-dire continue. Vérifier que la courbe est lisse aux points de raccord (au niveau des sommets du polygone) est plus subtil. La courbe $y_\theta(\theta)$ approxime la courbe $y(x)$ avec $x = x_0 + \theta\,h$. Il faut vérifier que la dérivée par rapport à x de y_θ est la même à droite de l'intervalle courant (quand $\theta = 1$) et à gauche de l'intervalle suivant (quand $\theta = 0$). Il faut donc voir θ comme une fonction de x et appliquer la formule pour la dérivée des fonctions composées :

$$\theta = \frac{x - x_0}{h}\,, \quad \frac{\mathrm{d}y_\theta}{\mathrm{d}x} = \frac{\mathrm{d}y_\theta}{\mathrm{d}\theta}\,\frac{\mathrm{d}\theta}{\mathrm{d}x} = \frac{1}{h}\,\frac{\mathrm{d}y_\theta}{\mathrm{d}\theta}.$$

En remplaçant y_1 par $y_0 + h\,(b_1\,k_1 + b_2\,k_2 + \cdots + b_s\,k_s)$ dans (4.22), on obtient :

$$\frac{\mathrm{d}y_\theta}{\mathrm{d}x} = 6\,\theta\,(1-\theta)\left(\sum_{i=1}^{s} b_i\,k_i\right) + \left(3\,\theta^2 - 4\,\theta + 1\right)\,f(x_0,y_0) + \left(3\,\theta^2 - 2\,\theta\right)\,f(x_1,y_1)\,. \tag{4.23}$$

À droite de l'intervalle courant, en $\theta = 1$, cette dérivée vaut $f(x_1, y_1)$. À gauche de l'intervalle suivant, en $\theta = 0$, elle vaut $f(x_0, y_0)$. Lors du changement d'intervalle, la nouvelle valeur de (x_0, y_0) est égale à l'ancienne valeur de (x_1, y_1). Les deux dérivées sont donc égales et la courbe est lisse, même en cas d'intégrateur à pas adaptatif, puisque ces expressions ne dépendent pas de h.

Autre façon d'obtenir la formule. Il est remarquable que le polynôme (4.22) est la solution d'un problème d'interpolation (voir section 1.5). Il suffit de poser

$$y_\theta = a_3\,\theta^3 + a_2\,\theta^2 + a_1\,\theta + a_0\,, \quad y_\theta(0) = y_0\,, \quad y_\theta(1) = y_1\,,$$

$$\frac{\mathrm{d}y_\theta}{\mathrm{d}\theta}(0) = h\,f(x_0,y_0)\,, \quad \frac{\mathrm{d}y_\theta}{\mathrm{d}\theta}(1) = h\,f(x_1,y_1)$$

et de déterminer les coefficients de y_θ en résolvant le système linéaire ainsi défini. Illustration en MAPLE :

```
> with (LinearAlgebra):
> A := <<1,1,0,0>|<0,1,1,1>|<0,1,0,2>|<0,1,0,3>>;

                                [1    0    0    0]
                                [                ]
                                [1    1    1    1]
                           A := [                ]
                                [0    1    0    0]
                                [                ]
                                [0    1    2    3]

> v := <y0, y1, h*f(x0,y0), h*f(x1,y1)>;

                                  [    y0     ]
                                  [           ]
                                  [    y1     ]
                             v := [           ]
                                  [h f(x0, y0)]
                                  [           ]
                                  [h f(x1, y1)]

> sol := LinearSolve (A, v):
> ytheta := add (sol[i]*theta^(i-1), i = 1..Dimension(sol));

ytheta := y0 + h f(x0, y0) theta

                                                          2
     + (-3 y0 - 2 h f(x0, y0) + 3 y1 - h f(x1, y1)) theta

                                                      3
     + (h f(x0, y0) + 2 y0 - 2 y1 + h f(x1, y1)) theta
```

La courbe donne une approximation d'ordre 3. Cette construction-ci [1, page 98] a l'intérêt de montrer pourquoi la formule (4.22) fournit une approximation d'ordre 3 de la solution exacte. On part de la formule qui donne y_1 et on place des étoiles partout (on a supposé pour simplifier que le nombre d'étages ne changeait pas mais on aurait pu généraliser) :

$$y_1^* \quad = \quad y_0 + h^* \sum_{i=1}^{s} b_i^* \, k_i^* \tag{4.24}$$

avec $h^* = \theta \, h$, afin d'approximer $y(x_0 + \theta \, h)$. Les k_i^* sont naturellement définis par :

$$k_i^* \quad = \quad f \left(x_0 + c_i^* \, h^*, \, y_0 + h^* \sum_{j=1}^{i-1} a_{ij}^* \, k_j^* \right) . \tag{4.25}$$

On simplifie beaucoup les choses en posant $c_i^* = c_i / \theta$ et $a_{ij}^* = a_{ij} / \theta$, parce qu'alors $k_i^* = k_i$. La formule (4.24) utilise donc les évaluations de fonction qui ont servi à calculer y_1. Elle se réécrit

en :

$$y_1^* \quad = \quad y_0 + h^* \sum_{i=1}^{s} b_i^*\, k_i\,. \tag{4.26}$$

Comment obtient-on les b_i^* ? En procédant comme en section 4.4.3, c'est-à-dire en posant que le développement en série de la solution exacte $y(x_0 + h^*)$ coïncide avec celui de la solution numérique $y_1^*(h^*)$ jusqu'au terme en $(h^*)^3$ inclus. Pour fixer les idées, on a pris l'exemple d'un schéma d'ordre $p = 3$ (le schéma de Heun), auquel on a ajouté un quatrième étage ($s = 4$) de façon à ce que $k_s = f(x_1, y_1)$. Le tableau de ce schéma est :

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{3} & \frac{1}{3} & & & \\ \frac{2}{3} & 0 & \frac{2}{3} & & \\ 1 & \frac{1}{4} & 0 & \frac{3}{4} & \\ \hline & \frac{1}{4} & 0 & \frac{3}{4} & 0 \end{array}$$

On a calculé la différence entre les deux développements (comme en section 4.4.3) et remplacé les coefficients c_i^* et a_{ij}^* par c_i/θ et a_{ij}/θ. Les conditions à satisfaire sont données par les coefficients qui doivent s'annuler identiquement :

$$b_1^* + b_2^* + b_3^* + b_4^* = 1\,, \quad \frac{2}{3}\,b_2^* + \frac{4}{3}\,b_3^* + 2\,b_4^* = \theta\,, \quad \frac{1}{3}\,b_2^* + \frac{4}{3}\,b_3^* + 3\,b_4^* = \theta^2\,, \quad \frac{4}{3}\,b_3^* + 3\,b_4^* = \theta^2\,.$$

Un calcul rapide aboutit à :

$$b_2^* = 0\,, \quad b_4^* = \theta^2 - \theta\,, \quad b_3^* = -\frac{3}{2}\,\theta^2 + \frac{9}{4}\,\theta\,, \quad b_1^* = \frac{1}{2}\,\theta^2 - \frac{5}{4}\,\theta + 1\,.$$

En remplaçant les b_i^* par leur valeur et h^* par $\theta\, h$ dans (4.26), on obtient une formule dépendant de θ et de h qui constitue une approximation de $y(x_0 + \theta\, h)$ d'ordre 3. Cette formule est égale à la formule (4.22). Pour un autre schéma, les polynômes b_i^* changent mais pas la formule (4.22).

Question 4. Refaire les calculs pour le schéma d'ordre 3 suivant [1, page 100] (ne pas oublier d'ajouter un étage).

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ \frac{3}{4} & 0 & \frac{3}{4} & \\ \hline & \frac{2}{9} & \frac{1}{3} & \frac{4}{9} \end{array}$$

Les polynômes à obtenir sont

$$b_1^* = 1 - \frac{4}{3}\,\theta + \frac{5}{9}\,\theta^2\,, \quad b_2^* = \theta\,\left(1 - \frac{2}{3}\,\theta\right)\,, \quad b_3^* = \frac{4}{3}\,\theta\left(1 - \frac{2}{3}\,\theta\right)\,, \quad b_4^* = \theta\,(\theta - 1)\,.$$

4.5.3 Implantation en FORTRAN

On conseille le découpage suivant : dans un fichier `integrateur.f`, un programme principal (chargé du choix de l'exemple à traiter, du schéma à utiliser et même de la nature de l'intégrateur), un sous-programme pour l'intégrateur à pas adaptatif (chargé de gérer les pas acceptés ou rejetés et de lisser la courbe) et un sous-programme pour calculer y_1 et $\hat{y}_1$ en fonction de y_0. Les deux fonctions qui définissent le problème à traiter peuvent être placées dans un fichier séparé, par exemple : `Arenstorf.f`. La fonction qui définit le schéma de Runge-Kutta à utiliser peut être placée dans un autre fichier séparé, par exemple : `Zonneveld.f`. Un programme principal élémentaire pour `integrateur.f` pourrait ressembler à ceci :

```
      PROGRAM DSOLVE
      IMPLICIT NONE
* La dimension maximale des systèmes à intégrer
      INTEGER NMAX
      PARAMETER (NMAX = 10)
*
* LES VARIABLES DU PROBLÈME DIFFÉRENTIEL
*
* N             La (vraie) dimension de Y
* X0, XEND      L'intervalle d'intégration
* PROBLEM_F     Sous-programme chargé de l'évaluation de
*               la fonction F associée au problème
      INTEGER N
      DOUBLE PRECISION X0, XEND, Y(NMAX)
      EXTERNAL PROBLEM_F
*
* LES VARIABLES DU SCHÉMA DE RUNGE-KUTTA
*
* SMAX          Le nombre maximal d'étages
* S             Le (vrai) nombre d'étages
* P             L'ordre de la formule principale
* PHAT          L'ordre de la formule emboîtée
*
* A, B, C       Les coefficients du  tableau
* BHAT          B alternatif pour la formule emboîtée
*
* K             Les K(I) des formules de Runge-Kutta
*               K est un tableau de S colonnes de dimension N
*
      INTEGER SMAX, S, P, PHAT
      PARAMETER (SMAX = 7)
      DOUBLE PRECISION C(SMAX)
      DOUBLE PRECISION A(SMAX,SMAX-1)
      DOUBLE PRECISION B(SMAX), BHAT(SMAX)
      DOUBLE PRECISION K(NMAX,SMAX)
*
* LES VARIABLES DE L'INTÉGRATEUR
*
* EPSLN         L'erreur souhaitée (à la fois locale et globale)
* HMIN          Le pas minimum à effectuer, pour la sortie dense,
*               sous réserve que P >= 3.
```

```
*
      DOUBLE PRECISION EPSLN, HMIN
*
* Initialise les variables du problème
      CALL PROBLEM_INIT (N, X0, XEND, Y)
* Initialise les variables du schéma de Runge-Kutta
      CALL SCHEMA_INIT (S, P, PHAT, C, A, SMAX, B, BHAT)
      EPSLN = 1D-5
      HMIN = 1D-3
* Appelle l'intégrateur
      CALL RK_ADAPTATIF (N, X0, XEND, Y, PROBLEM_F,
     $                   S, P, PHAT, C, A, SMAX, B, BHAT, K, NMAX,
     $                   HMIN, EPSLN)
      END PROGRAM
```

Voici l'entête du sous-programme qui implante l'intégrateur. Les paramètres `LDA` et `LDK` contiennent le nombre de lignes des matrices `A` et `K`, telles qu'elles ont été déclarées (même convention qu'avec LAPACK).

```
      SUBROUTINE RK_ADAPTATIF (N, X0, XEND, Y0, FCN,
     $                   S, P, PHAT, C, A, LDA, B, BHAT, K, LDK,
     $                   HMIN, EPSLN)
      IMPLICIT NONE
* Le problème différentiel
      INTEGER N
      DOUBLE PRECISION X0, XEND, Y0(N)
      EXTERNAL FCN
* Le schéma de Runge-Kutta
      INTEGER S, P, PHAT, LDA, LDK
      DOUBLE PRECISION C(S)
      DOUBLE PRECISION A(LDA,S-1)
      DOUBLE PRECISION B(S), BHAT(S)
*              K est un tableau de S colonnes de dimension N
      DOUBLE PRECISION K(LDK,S)
* L'intégrateur
      DOUBLE PRECISION HMIN, EPSLN
*
* VARIABLES LOCALES POUR LA GESTION DE L'ERREUR LOCALE
*
      DOUBLE PRECISION H, HNEW, ERREUR
*
* X1            La nouvelle valeur de X0 (si le pas est accepté)
* Y1            La nouvelle valeur de Y0 (si le pas est accepté)
* Y1HAT         Y1 alternatif, produit par la formule emboîtée
*
      DOUBLE PRECISION X1, Y1(N), Y1HAT(N)
```

Certaines BLAS peuvent être utiles : `DCOPY` (copie de vecteurs), `DSCAL` (multiplication par un scalaire, mise à zéro d'un vecteur), `DAXPY` (combinaison linéaire $\alpha\, x + y$ de deux vecteurs). Le sous-programme suivant initialise les variables associées à un schéma de Runge-Kutta. Voici l'exemple du schéma de Zonneveld 4(3). Il est possible de le placer dans un fichier séparé de l'intégrateur, par exemple `Zonneveld.f`, pour faciliter l'utilisation de différents schémas.

```
      SUBROUTINE SCHEMA_INIT (S, P, PHAT, C, A, LDA, B, BHAT)
* Le schéma Zonneveld 4(3)
      IMPLICIT NONE
      INTEGER S, P, PHAT, LDA
      DOUBLE PRECISION C(*)
      DOUBLE PRECISION A(LDA,*)
      DOUBLE PRECISION B(*), BHAT(*)
*
      S = 5
      P = 4
      PHAT = 3
*
      C(1) = 0D0
      C(2) = 5D-1
      C(3) = 5D-1
      C(4) = 1D0
      C(5) = 75D-2
      A(2,1) = 5D-1
      A(3,1) = 0D0
      A(3,2) = 5D-1
      A(4,1) = 0D0
      A(4,2) = 0D0
      A(4,3) = 1D0
      A(5,1) = 5D0/32D0
      A(5,2) = 7D0/32D0
      A(5,3) = 13D0/32D0
      A(5,4) = -1D0/32D0
      B(1) = 1D0/6D0
      B(2) = 1D0/3D0
      B(3) = 1D0/3D0
      B(4) = 1D0/6D0
      B(5) = 0D0
* Formule emboîtée
      BHAT(1) = -5D-1
      BHAT(2) = 7D0/3D0
      BHAT(3) = 7D0/3D0
      BHAT(4) = 13D0/6D0
      BHAT(5) = -16D0/3D0
      END SUBROUTINE
```

Les sous-programmes suivants peuvent être utilisés pour le calcul des orbites d'Arenstorf. Il est possible de placer ces sous-programmes dans un fichier séparé de l'intégrateur, par exemple `Arenstorf.f`, pour faciliter le traitement de différents exemples.

```
      SUBROUTINE PROBLEM_INIT (N, X0, XEND, Y0)
*
* Des conditions initiales pour Arenstorf (Hairer, Norsett, Wanner, p. 130)
*
* Y = (y1, y2, y1', y2')
*
      IMPLICIT NONE
      INTEGER N
```

```
      DOUBLE PRECISION X0, XEND, Y0(*)
*
      X0 = 0D0
      XEND = 17.06521656015796255889172062 49
      N = 4
      Y0(1) = 0.994D0
      Y0(2) = 0D0
      Y0(3) = 0D0
      Y0(4) = -2.00158510637908252240537862224D0
      END SUBROUTINE

      SUBROUTINE PROBLEM_F (N, X, Y, F)
*
* Une fonction R x R^N -> R^n
*              (X, Y)  -> F = FCN (X, Y)
*
* Les vecteurs Y et F sont de dimension N = 4
* Il s'agit d'un système dans le plan (y1, y2) d'ordre 2.
*
* Y = (y1, y2, y1', y2')
*
* Voir Hairer, Norsett, Wanner, p. 476 pour le code FORTRAN
* et p. 129 pour le système mathématique
*
      IMPLICIT NONE
      INTEGER N
      DOUBLE PRECISION X, Y(N), F(N)
*
      DOUBLE PRECISION MU, MUP, R1, R2
      PARAMETER (MU = 0.012277471D0)
      PARAMETER (MUP = 1D0 - MU)
      F(1) = Y(3)
      F(2) = Y(4)
      R1 = (Y(1) + MU)**2+Y(2)**2
      R1 = R1 * SQRT(R1)
      R2 = (Y(1) - MUP)**2+Y(2)**2
      R2 = R2 * SQRT(R2)
      F(3) = Y(1)+2D0*Y(4)-MUP*(Y(1)+MU)/R1-MU*(Y(1)-MUP)/R2
      F(4) = Y(2)-2D0*Y(3)-MUP*Y(2)/R1-MU*Y(2)/R2
      END SUBROUTINE
```

4.6 Les schémas de Runge-Kutta implicites

Cette section joue davantage le rôle d'une conclusion du chapitre 4 que celui d'une véritable présentation des intégrateurs implicites. Au delà de son intérêt scientifique, elle permet de comprendre pourquoi les bibliothèques logicielles dédiées à l'intégration numérique des équations différentielles demandent parfois qu'on leur fournisse non seulement la fonction f qui définit le système différentiel mais aussi sa matrice jacobienne.

4.6.1 Motivation

On commence par deux exemples de simulations numériques, qui comparent les performances des méthodes explicites décrites dans les sections précédentes avec une méthode implicite. Sur le fond, les méthodes implicites sont meilleures sur de tels systèmes parce qu'elles ont de meilleures propriétés de stabilité [3, chapter IV.2].

Le système de Robertson. Il s'agit d'un système de réactions chimiques abstrait comportant trois espèces chimiques A, B et C ainsi que trois réactions. Les réactions sont étiquetées par des constantes cinétiques. Ces constantes ont des ordres de grandeur très différents. On a affaire à une réaction lente, une rapide et une très rapide : c'est ce qui rend le système difficile à intégrer. Voir la Figure 4.10 ainsi que [3, chapter IV.1, page 3].

$$A \xrightarrow{4\,10^{-2}} B\,, \quad B + C \xrightarrow{10^4} A + C\,, \quad 2\,B \xrightarrow{3\,10^7} B + C\,.$$

Les concentrations de A, B et C sont notées $y_1(x)$, $y_2(x)$ et $y_3(x)$ ($n = 3$). Le système différentiel s'écrit

$$\begin{aligned} \dot{y}_1 &= -4\,10^{-2}\,y_1 + 10^4\,y_2\,y_3\,, \\ \dot{y}_2 &= 4\,10^{-2}\,y_1 - 10^4\,y_2\,y_3 - 3\,10^7\,y_2^2\,, \\ \dot{y}_3 &= 3\,10^7\,y_2^2\,. \end{aligned}$$

avec les conditions initiales suivantes :

$$y_1(0) = 1\,, \quad y_2(0) = y_3(0) = 0\,.$$

L'équation de Van Der Pol. Il s'agit d'une équation différentielle d'ordre deux, qui modélise le comportement d'un circuit électrique ($n = 2$). Ici aussi, c'est le très petit ordre de grandeur de μ qui rend le système difficile à intégrer. Voir la Figure 4.11 [3, chapter IV.1, page 4].

$$\begin{aligned} \dot{y}_1 &= y_2\,, \\ \dot{y}_2 &= \mu\,((1 - y_1^2)\,y_2 - y_1)\,. \end{aligned}$$

avec

$$\mu = 10^{-6}\,, \quad y_1(0) = 2\,, \quad y_2(0) = 6\,10^{-1}.$$

4.6.2 Structure des schémas

Les schémas implicites sont caractérisés par le fait que la matrice des a_{ij} est carrée.

$$\begin{array}{c|ccc} c_1 & a_{11} & \cdots & a_{1s} \\ \vdots & \vdots & & \vdots \\ c_s & a_{s1} & \cdots & a_{ss} \\ \hline & b_1 & \cdots & b_s \end{array}$$

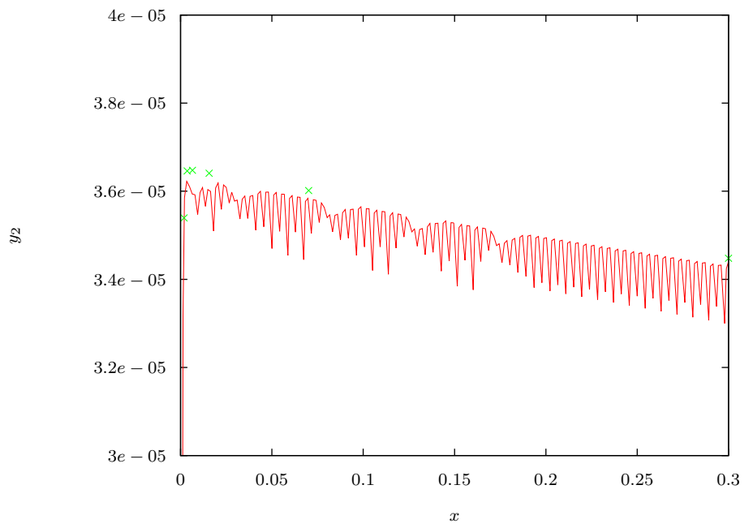

FIGURE 4.10 – Simulation numérique du système de Robertson. La courbe rouge a été produite par l'intégrateur à pas adaptatif décrit en section 4.5, muni du schéma Dormand-Prince 5(4) avec $\varepsilon = 10^{-6}$. L'intégrateur accepte 208 pas et en rejette 83. Par comparaison, les croix représentent les 7 points calculés par un intégrateur à pas adaptatif, muni du schéma de Runge-Kutta implicite Radau IIA d'ordre 5(4) avec la même valeur de ε. La vraie solution n'oscille pas et passe très près des croix. Il est manifeste que la méthode implicite se comporte beaucoup mieux que la méthode explicite.

Voici par exemple le tableau associé au schéma RadauIIA à $s = 3$ étages, d'ordre $p = 5$ [3, chapter IV, page 74] :

$\frac{4-\sqrt{6}}{10}$	$\frac{88-7\sqrt{6}}{360}$	$\frac{296-169\sqrt{6}}{1800}$	$\frac{-2+3\sqrt{6}}{225}$
$\frac{4+\sqrt{6}}{10}$	$\frac{296+169\sqrt{6}}{1800}$	$\frac{88+7\sqrt{6}}{360}$	$\frac{-2-3\sqrt{6}}{225}$
1	$\frac{16-\sqrt{6}}{36}$	$\frac{16+\sqrt{6}}{36}$	$\frac{1}{9}$
	$\frac{16-\sqrt{6}}{36}$	$\frac{16+\sqrt{6}}{36}$	$\frac{1}{9}$

4.6.3 Le système non linéaire à résoudre

La première difficulté tient au fait que les k_i ne peuvent plus se calculer les uns après les autres par une simple évaluation mais sont la solution d'un système non linéaire, qui devra être résolu à chaque pas de l'intégrateur.

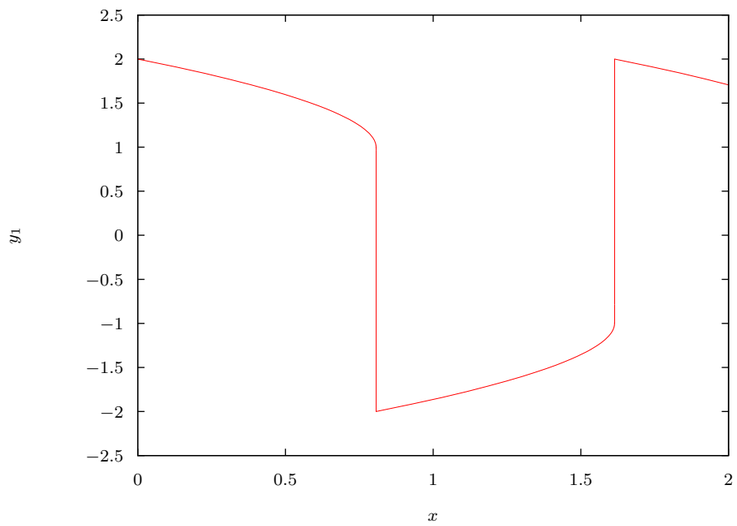

FIGURE 4.11 – Simulation numérique de l'équation de Van Der Pol. Les deux intégrateurs donnent visuellement la même courbe, mais pas avec le même travail. La méthode implicite accepte 865 pas et en rejette 13. La méthode explicite accepte 1160503 pas et en rejette 29272.

Les formules du système (4.7) deviennent :

$$\begin{aligned} k_i &= f(x_0 + c_i\,h,\, y_0 + \underbrace{h\,(a_{i1}\,k_1 + a_{i2}\,k_2 + \cdots + a_{is}\,k_s)}_{z_i}) \quad (1 \leqslant i \leqslant s)\,, \\ y_1 &= y_0 + h\,(b_1\,k_1 + b_2\,k_2 + \cdots + b_s\,k_s)\,. \end{aligned} \tag{4.27}$$

Tant qu'à résoudre un système non linéaire, autant prendre pour variables les expressions z_i en entrée des fonctions f (on préfère prendre les z_i pour variables plutôt que les quantités $g_i = y_0 + z_i$ pour éviter des problèmes d'arrondis). On reformule donc le système (4.27) en le système équivalent suivant [3, chapter IV.3, page 40 et IV.8, page 118] :

$$z_i = h \sum_{j=1}^{s} a_{ij}\, f(x_0 + c_j\,h,\, y_0 + z_j) \quad (1 \leqslant i \leqslant s)\,, \tag{4.28}$$

$$y_1 = y_0 + h \sum_{j=1}^{s} b_j\, f(x_0 + c_j\,h,\, y_0 + z_j)\,. \tag{4.29}$$

Le système (4.28) comporte s variables z_i qui sont elles-mêmes des vecteurs de dimension n. On a donc affaire à un système de $n \times s$ équations non linéaires en $n \times s$ variables réelles. Ce système se résout par une méthode de Newton, qui utilise la matrice jacobienne J_z du système (4.28).

Cette grosse matrice jacobienne de dimension $(n \times s) \times (n \times s)$ se construit elle-même à partir de la matrice jacobienne J_f, de dimension $n \times n$ de la fonction f. C'est cette matrice J_f que les bibliothèques logicielles demandent qu'on leur fournisse, lorsqu'on souhaite utiliser une méthode implicite.

Rappelons que la fonction f est un vecteur de n fonctions $f_i(x, y_1, y_2, \ldots, y_n)$. La matrice jacobienne J_f est elle-même une fonction $J_f(x, y_1, y_2, \ldots, y_n)$. Elle s'écrit :

$$J_f = \begin{pmatrix} \frac{\partial f_1}{\partial y_1} & \cdots & \frac{\partial f_1}{\partial y_n} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial y_1} & \cdots & \frac{\partial f_n}{\partial y_n} \end{pmatrix}.$$

La matrice jacobienne J_z peut alors s'écrire :

$$J_z = \begin{pmatrix} I - h\,a_{11}\,J_f(x_0 + c_1\,h,\, y_0 + z_1) & \cdots & h\,a_{1s}\,J_f(x_0 + c_s\,h,\, y_0 + z_s) \\ \vdots & & \vdots \\ h\,a_{s1}\,J_f(x_0 + c_1\,h,\, y_0 + z_1) & \cdots & I - h\,a_{ss}\,J_f(x_0 + c_s\,h,\, y_0 + z_s) \end{pmatrix}.$$

L'itération de Newton est donnée par (4.31).

4.6.4 Méthode de Newton

Dans cette section, on rappelle le principe de la méthode de Newton en une et en plusieurs variables. On montre ensuite comment l'appliquer au système (4.28).

Cas d'une variable. On suppose qu'on connaît une approximation $x^{(0)}$ d'une solution d'une équation $f(x) = 0$, où f est une fonction dérivable. La méthode de Newton calcule une suite de réels qui converge (sous certaines hypothèses) vers cette solution. On considère le début d'un développement en série, centré en $x^{(0)}$, de la fonction f :

$$f(x^{(0)} + h) \quad \simeq \quad f(x^{(0)}) + h\,f'(x^{(0)}) \tag{4.30}$$

On cherche h tel que $f(x^{(0)} + h) = 0$. On trouve

$$h \quad = \quad -\frac{f(x^{(0)})}{f'(x^{(0)})}, \qquad x^{(0)} + h \quad = \quad x^{(0)} - \frac{f(x^{(0)})}{f'(x^{(0)})}\cdot$$

Comme le développement en série (4.30) est tronqué, le nombre $x^{(0)} + h$ trouvé n'est qu'une approximation de la racine recherchée et on recommence la calcul en le prenant comme nouveau point de départ. En général, si la suite converge vers la racine visée, elle converge vite [18]. La formule générale s'écrit :

$$x^{(i+1)} \quad = \quad x^{(i)} - \frac{f(x^{(i)})}{f'(x^{(i)})}\cdot$$

18. La vitesse de convergence est quadratique en général, c'est-à-dire que le nombre de décimales exactes double à chaque itération.

Cas de plusieurs variables. On considère une fonction dérivable $F : \mathbb{R}^2 \to \mathbb{R}^2$. La fonction F est donc un vecteur $F = \begin{pmatrix} f(x,y) & g(x,y) \end{pmatrix}^T$ de deux fonctions de deux variables réelles x et y. On suppose qu'on connaît une approximation $\begin{pmatrix} x^{(0)} & y^{(0)} \end{pmatrix}^T$ d'une solution de l'équation $F(x,y) = 0$. La méthode de Newton calcule une suite de vecteurs de $\mathbb{R}^2$ qui converge (sous certaines hypothèses) vers cette solution. On considère le début d'un développement en série, centré en $\begin{pmatrix} x^{(0)} & y^{(0)} \end{pmatrix}^T$, de la fonction F :

$$\begin{aligned} f(x^{(0)}+h,\, y^{(0)}+\ell) &= f(x^{(0)},y^{(0)}) + h\,\frac{\partial f}{\partial x}(x^{(0)},y^{(0)}) + \ell\,\frac{\partial f}{\partial y}(x^{(0)},y^{(0)})\,, \\ g(x^{(0)}+h,\, y^{(0)}+\ell) &= g(x^{(0)},y^{(0)}) + h\,\frac{\partial g}{\partial x}(x^{(0)},y^{(0)}) + \ell\,\frac{\partial g}{\partial y}(x^{(0)},y^{(0)})\,. \end{aligned}$$

Ce début de développement en série peut s'écrire matriciellement. On voit alors apparaître la matrice jacobienne $J(x,y)$ de la fonction $F(x,y)$:

$$\begin{pmatrix} f(x^{(0)}+h,\, y^{(0)}+\ell) \\ g(x^{(0)}+h,\, y^{(0)}+\ell) \end{pmatrix} = \begin{pmatrix} f(x^{(0)},y^{(0)}) \\ g(x^{(0)},y^{(0)}) \end{pmatrix} + \underbrace{\begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{pmatrix}}_{J}(x^{(0)},y^{(0)}) \cdot \begin{pmatrix} h \\ \ell \end{pmatrix}.$$

Comme dans le cas d'une variable, on pose que le membre gauche de l'équation précédente est nul. Les inconnues, qui sont les accroissements h et ℓ, sont donc solution du système d'équations linéaires suivant :

$$J(x^{(0)},y^{(0)}) \cdot \begin{pmatrix} h \\ \ell \end{pmatrix} = -F(x^{(0)},y^{(0)})\,.$$

L'itération de Newton est donc définie par :

$$\begin{aligned} J(x^{(i)},y^{(i)}) \cdot \begin{pmatrix} h^{(i)} \\ \ell^{(i)} \end{pmatrix} &= -F(x^{(i)},y^{(i)}) \quad \text{(à résoudre)}\,, \\ \begin{pmatrix} x^{(i+1)} \\ y^{(i+1)} \end{pmatrix} &= \begin{pmatrix} x^{(i)} \\ y^{(i)} \end{pmatrix} + \begin{pmatrix} h^{(i)} \\ \ell^{(i)} \end{pmatrix} \end{aligned}$$

Lien avec la section précédente. Les variables z_j s'obtiennent en répétant un certain nombre de fois l'itération suivante, dont les inconnues sont les accroissements Δz_j (comme les z_j, ce sont s vecteurs de dimension n). Voir [3, chapter IV.8, pages 118-120].

$$\begin{aligned} J_z(z_1^{(i)},\ldots,z_s^{(i)}) \cdot \begin{pmatrix} \Delta z_1^{(i)} \\ \vdots \\ \Delta z_s^{(i)} \end{pmatrix} &= -\begin{pmatrix} f(x_0+c_1\,h,\, y_0+z_1^{(i)}) \\ \vdots \\ f(x_0+c_s\,h,\, y_0+z_s^{(i)}) \end{pmatrix} \quad \text{(à résoudre)}\,, \\ \begin{pmatrix} z_1^{(i+1)} \\ \vdots \\ z_s^{(i+1)} \end{pmatrix} &= \begin{pmatrix} z_1^{(i)} \\ \vdots \\ z_s^{(i)} \end{pmatrix} + \begin{pmatrix} \Delta z_1^{(i)} \\ \vdots \\ \Delta z_s^{(i)} \end{pmatrix} \end{aligned} \tag{4.31}$$

Il est souvent conseillé [3, chapter IV.8, page 119] d'approximer la matrice $J_f(x_0 + c_i\,h,\, y_0 + z_i)$ par la matrice $J_f(x_0, y_0)$. La matrice J_z reste alors la même à chaque itération. Sa factorisation $L\,U$ peut n'être calculée qu'une seule fois et être réutilisée à chaque résolution de système linéaire. Une implantation, prévue dans le cadre d'un intégrateur à pas adaptatif, est donnée Figure **??**.

4.6.5 Détermination de l'ordre

Comment détermine-t-on l'ordre p d'un schéma implicite ? Sur le principe, l'idée est la même que pour les schémas explicites : on compare le développement en série de la solution exacte $y(x_0 + h)$ et celui de $y_1(h)$. Les deux séries doivent être égales terme à terme jusqu'au coefficient de h^p inclus. En pratique, le problème du développement en série de $y_1(h)$ peut laisser perplexe ! En effet, dans le cas explicite, le développement en série a été calculé à partir de la formule qui donne y_1, en remplaçant tous les k_i par leur valeur. Mais dans le cas implicite, il n'est pas possible de remplacer les k_i par leur valeur puisqu'ils dépendent tous les uns des autres ! En fait, la difficulté n'est qu'apparente : il suffit d'effectuer les remplacements paresseusement (voir toutefois [2, chapter II.7, pages 206-207] pour la question de l'existence d'une solution numérique). Voici une illustration du procédé en MAPLE. On commence comme dans le cas explicite :

```
> edo := diff (y(x),x) = f(x,y(x));
                                d
                      edo := -- y(x) = f(x, y(x))
                                dx
> s := 3:
> p := 5:
```

Les commandes suivantes calculent le développement en série de la solution exacte, jusqu'à l'ordre $p + 1$. Comme l'expression résultat est très volumineuse, on tronque la sortie.

```
> vsol := dsolve ({edo, y(x0)=y0}, y(x), series, order=p+2):
> vsol := collect (subs (x=x0+h, convert (rhs (vsol), polynom)), h);

vsol := y0 + f(x0, y0) h

                                                                  2
     + (1/2 D[1](f)(x0, y0) + 1/2 D[2](f)(x0, y0) f(x0, y0)) h  + [...]
```

Pour la solution numérique y_1, l'idée consiste à représenter les k_j par des fonctions inconnues $k_j(h)$ et à introduire séparément, pour chaque k_j, une expression $ek(j)$ qui servira ultérieurement à remplacer la fonction inconnue $k_j(h)$ par sa valeur, développée une étape plus loin.

```
> ek := j -> f(x0 + c[j]*h, y0 + h*add (a[j,l]*k[l](h), l = 1..s));

      ek := j -> f(x0 + c[j] h, y0 + h add(a[j, l] k[l](h), l = 1 .. s))
```

```
> y1 := y0 + h*add(b[i]*f(x0 + c[i]*h,
                          y0 + h*add (a[i,j]*k[j](h), j = 1..s)), i = 1..s);

y1 := y0 + h (b[1] f(x0 + c[1] h,

    y0 + h (a[1, 1] k[1](h) + a[1, 2] k[2](h) + a[1, 3] k[3](h))) + b[2] f(

    x0 + c[2] h, y0 + h (a[2, 1] k[1](h) + a[2, 2] k[2](h) + a[2, 3] k[3](h)))

     + b[3] f(x0 + c[3] h,

    y0 + h (a[3, 1] k[1](h) + a[3, 2] k[2](h) + a[3, 3] k[3](h))))
```

Il ne reste plus qu'à définir la différence entre les deux expressions (il est inutile de développer y_1 en série puisque les calculs menés ci-dessous le feront automatiquement) :

```
> difference := y1 - vsol:
> X := indets (difference, function):
```

On stocke maintenant les coefficients qui définissent le schéma de la section 4.6.2 dans une variable.

```
> RadauIIA :=
>         c[1]=(4-sqrt(6))/10, c[2]=(4+sqrt(6))/10, c[3]=1,
>         a[1,1]=(88-7*sqrt(6))/360,
>         a[1,2]=(296-169*sqrt(6))/1800,
>         a[1,3]=(-2+3*sqrt(6))/225,
>         a[2,1]=(296+169*sqrt(6))/1800,
>         a[2,2]=(88+7*sqrt(6))/360,
>         a[2,3]=(-2-3*sqrt(6))/225,
>         a[3,1]=(16-sqrt(6))/36, a[3,2]=(16+sqrt(6))/36, a[3,3]=1/9,
>         b[1]=(16-sqrt(6))/36, b[2]=(16+sqrt(6))/36, b[3]=1/9:
```

Les deux développements en série commencent par le même terme, puisque leur différence vaut zéro :

```
> eval (difference, h = 0);
                                        0
```

Le coefficient de h dans la différence s'obtient en dérivant cette expression par rapport à h et en évaluant le résultat en zéro :

```
> collect (eval (diff (difference, h), h = 0), X);
                      (b[1] + b[2] + b[3] - 1) f(x0, y0)
```

On vérifie que le schéma satisfait bien à $b_1 + b_2 + b_3 = 1$:

```
> subs (RadauIIA, %);
                                        0
```

Pour obtenir le coefficient de h^2, on doit remplacer les fonctions inconnues $k_j(h)$ par $ek(j)$ une première fois, dériver deux fois par rapport à h puis évaluer en $h = 0$:

```
> difference := subs (seq (k[z](h) = ek(z), z = 1 .. s), difference):
> collect (eval (diff (difference, h, h), h = 0), X);
(2 b[1] (a[1, 1] + a[1, 2] + a[1, 3]) + 2 b[2] (a[2, 1] + a[2, 2] + a[2, 3])

     + 2 b[3] (a[3, 1] + a[3, 2] + a[3, 3]) - 1) D[2](f)(x0, y0) f(x0, y0)

     + (2 b[1] c[1] + 2 b[2] c[2] + 2 b[3] c[3] - 1) D[1](f)(x0, y0)
```

On vérifie que le schéma considéré annule identiquement cette expression :

```
> simplify (subs (RadauIIA, %));
                                        0
```

En continuant de proche en proche, il est possible de vérifier (avec une bonne machine, parce que les expressions grossissent très vite) que le schéma donné en section 4.6.2 est d'ordre $p = 5$.

4.6.6 Intégrateur à pas adaptatif

Sur le principe, l'idée est la même que pour les schémas explicites : on estime l'erreur en utilisant une formule emboîtée d'ordre $\hat{p} \neq p$ puis on avance et on augmente le pas si l'erreur est plus petite qu'un $\varepsilon > 0$ fixé à l'avance, sinon on rejette le pas et on diminue sa longueur. En pratique, les calculs sont plus techniques et l'algorithme de l'intégrateur peut même dépendre de certaines propriétés des schémas. On donne ici l'algorithme de contrôle du pas qui est exposé dans [3, chapter IV.8, pages 123-124]. Cet algorithme suppose que la matrice (a_{ij}) admet au moins une valeur propre réelle.

On commence par munir le schéma RadauIIA à 3 étages donné en section 4.6.2 de coefficients $\hat{b}_i$ définissant une formule emboîtée d'ordre $\hat{p} = 3$:

$$\begin{pmatrix} \hat{b}_1 & \hat{b}_2 & \hat{b}_3 \end{pmatrix} \;=\; \begin{pmatrix} b_1 & b_2 & b_3 \end{pmatrix} + \frac{\gamma_0}{3} \begin{pmatrix} -13 - 7\sqrt{6} & -13 + 7\sqrt{6} & -1 \end{pmatrix}$$

où

$$\gamma_0 \;=\; \frac{3 \times 3^{\frac{1}{3}} - 3^{\frac{2}{3}} + 6}{30}$$

désigne la valeur propre réelle de la matrice (a_{ij}) de ce schéma. Voir [3, chapter IV.8, (8.17) et (8.18), page 123]. La formule choisie pour $\hat{y}_1$ est [3, chapter IV.8, (8.16)] :

$$\hat{y}_1 \;=\; y_0 + h \left(\gamma_0 \, f(x_0, y_0) + \sum_{i=1}^{3} \hat{b}_i \, f(x_0 + c_i \, h, \, y_0 + z_i) \right) \tag{4.32}$$

On remarque qu'il est commode, d'un point de vue calculatoire, de calculer la différence $\hat{y}_1 - y_1$ plutôt que $\hat{y}_1$ (il vaut alors mieux mémoriser les coefficients $e_i = \hat{b}_i - b_i$ plutôt que les $\hat{b}_i$ et adapter le code de la Figure **??**). En effet, cette différence s'exprime simplement à partir des variables z_i [3, chapter IV.8, (8.17)] :

$$\hat{y}_1 - y_1 \;=\; \gamma_0 \, h \, f(x_0, y_0) + e_1 \, z_1 + e_2 \, z_2 + e_3 \, z_3 \, .$$

Comme dans le cas explicite, la formule précédente est un peu trop simple et demande à être raffinée. On commence par résoudre

$$(I - h\,\gamma_0\,J_f(x_0, y_0))\;err^{(0)} \quad = \quad \hat{y}_1 - y_1\,.$$

La matrice J_f désigne la jacobienne de la fonction f. Le vecteur des inconnues, $err^{(0)}$, est de dimension n. Au premier pas de l'intégrateur et après chaque pas rejeté, on raffine encore davantage et on résout :

$$(I - h\,\gamma_0\,J_f(x_0, y_0))\;err^{(1)} \quad = \quad \gamma_0\,h\,f(x_0,\,y_0 + err^{(0)}) + e_1\,z_1 + e_2\,z_2 + e_3\,z_3\,.$$

On peut alors définir

$$erreur \quad = \quad \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{err_i^{(1)}}{(1 + \max(|y_{0i}|, |y_{1i}|))\,\varepsilon}\right)^2} \tag{4.33}$$

Il faut bien sûr utiliser $err^{(0)}$ si $err^{(1)}$ n'a pas été calculé. Voir [3, chap IV.8, (8.19), (8.20) et (8.21), pages 123-124] pour une justification de ces calculs. Le code d'un intégrateur à pas adaptatif synthétisant toutes ces idées est donné Figure **??**.

,,,,

,,,,,,,,

Bibliographie

[1] John R. Dormand. *Numerical Methods for Differential Equations : A Computational Approach (Engineering Mathematics)*, volume 3 of *Engineering Mathematics*. CRC Press, 1996.

[2] Ernst Hairer, Syvert Paul Norsett, and Gerhard Wanner. *Solving ordinary differential equations I. Nonstiff problems*, volume 8 of *Springer Series in Computational Mathematics*. Springer–Verlag, New York, 2nd edition, 1993.

[3] Ernst Hairer and Gerhard Wanner. *Solving ordinary differential equations II. Stiff and Differential–Algebraic Problems*, volume 14 of *Springer Series in Computational Mathematics*. Springer–Verlag, New York, 2nd edition, 1996.

[4] Ernst Hairer and Gerhard Wanner. Introduction à l'Analyse Numérique. Accessible sur `http://www.unige.ch/~hairer`, juin 2005.

[5] Jack K. Hale and Hüseyin Koçak. *Dynamics and Bifurcations*, volume 3 of *Texts in Applied Mathematics*. Springer–Verlag, New York, 1991.

[6] Edmund Whittaker. *A Treatise on the Analytical Dynamics of Particles and Rigid Bodies*. Cambridge University Press, fourth edition, 1965.

[7] Wikipedia. Richard Arenstorf. `https://en.wikipedia.org/wiki/Richard_Arenstorf`.

Table des figures

Index