Chapitre III

Équations différentielles ordinaires

Un problème d'intégration, traité en Chap. I et déjà difficile, consiste à trouver une fonction $y(x)$, si une fonction $f(x)$ est donnée, tel que

$$y' = f(x) \qquad \text{ou} \qquad y(x) = y_0 + \int_{x_0}^{x} f(t)\,dt\,.$$

Pour un *problème d'équation différentielle* nous cherchons une fonction $y(x)$ tel que

$$y' = f(x,y), \quad y(x_0) = y_0 \qquad \text{ou} \qquad y(x) = y_0 + \int_{x_0}^{x} f(t,y(t))\,dt \tag{0.1}$$

où, cette-fois-ci, la fonction donnée $f(x,y)$ dépend de x *et de* y. On peut ausssi avoir des *équations différentielles d'ordre supérieur*, comme par exemple

$$y'' = f(x,y,y'), \quad y(x_0) = y_0,\ y'(x_0) = v_0 \qquad \text{ou} \qquad \begin{array}{ll} y' = v, & y(x_0) = y_0, \\ v' = f(x,y,v), & v(x_0) = v_0 \end{array} \tag{0.2}$$

où on a écrit $v(x)$ à la place de $y'(x)$. Ceci est un cas spécial *d'un système d'équations différentielles*

$$\begin{array}{ll} y_1' = f_1(x,y_1,y_2), & y_1(x_0) = y_{10}, \\ y_2' = f_2(x,y_1,y_2), & y_2(x_0) = y_{20} \end{array} \qquad \text{ou} \qquad y' = f(x,y), \quad y(x_0) = y_0 \tag{0.3}$$

en notation vectorielle. On peut aussi avoir des systèmes de trois, quatre etc. équations. On peut encore écrire $x = y_0(x)$ avec $y_0' = 1$ et rajouter ceci comme équation supplémentaire au système. Ce système devient alors *un système autonome*

$$\begin{array}{lll} y_0' = 1 & & y_0(x_0) = x_0, \\ y_1' = f_1(y_0,y_1,y_2), & y_1(x_0) = y_{10}, & \\ y_2' = f_2(y_0,y_1,y_2), & y_2(x_0) = y_{20} & \end{array} \qquad \text{ou} \qquad y' = f(y), \quad y(x_0) = y_0\,. \tag{0.4}$$

De passer entre ces diverses notations et analogies, va nous guider dans la recherche et dans l'analyse des méthodes numériques pour ces problèmes.

Bibliographie sur ce chapitre

J.C. Butcher (1987): *The Numerical Analysis of Ordinary Differential Equations*. John Wiley & Sons. [MA 65/276]

M. Crouzeix & A.L. Mignot (1984): *Analyse Numérique des Equations Différentielles.* Masson. [MA 65/217]

E. Hairer, S.P. Nørsett & G. Wanner (1993): *Solving Ordinary Differential Equations I. Nonstiff Problems*. Springer Series in Comput. Math., vol. 8, 2nd edition. [MA 65/245]

E. Hairer & G. Wanner (1996): *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*. Springer Series in Comput. Math., vol. 14, 2nd edition. [MA 65/245]

E. Hairer, C. Lubich & G. Wanner (2002, 2006): *Geometric Numerical Integration ; Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer Series in Comput. Math., vol. 31. [MA 65/448], [OBSA 2594], [PHYA 5140]',; ces trois derniers livres "are often described as the "bibles" of their fields" (SIAM News, Philadelphia Dec. 2003)

P. Henrici (1962): *Discrete Variable Methods in Ordinary Differential Equations*. John Wiley & Sons. [MA 65/50]

III.1 Exemples d'équations différentielles

Un problème mécanique ; un point qui glisse sur une courbe.

Problème. On veut connaître le mouvement d'un point de masse glissant, sous effet de la gravité et sans frottement, sur une courbe donnée, par exemple $y = (1-x^2)^2$ (voir Fig. III.1, à gauche).

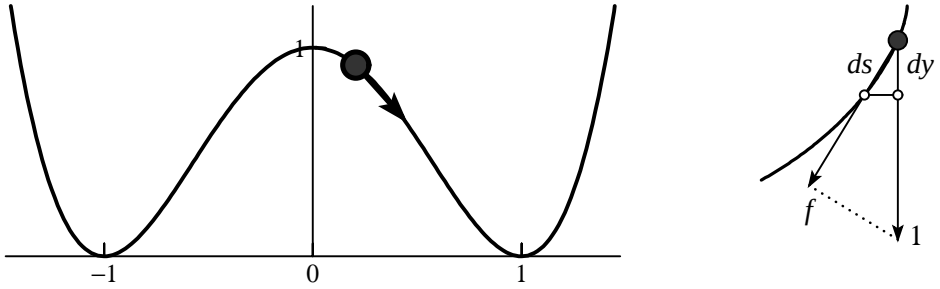


FIG. III.1: Problème mécanique ; un corps glissant dans une double vallée

Solution. Malgré le fait que tout enfant faisant de la luge connaît le problème, la solution analytique est assez difficile. Prenons la longueur d'arc s, avec $ds = \sqrt{dx^2+dy^2} = \sqrt{1+p^2}\,dx$ où $p = \frac{dy}{dx}$, comme coordonnée pour déterminer la position du corps. La force accélératrice f est, par Thales, $f = mg \cdot \frac{dy}{ds}$ (voir Fig. III.1, à droite), où nous posons pour la masse et la constante de gravitation $m = 1$ et $g = 1$. Ainsi $f = \frac{dy}{ds} = \frac{dy/dx}{ds/dx} = \frac{p}{\sqrt{1+p^2}}$. Avec la vitesse $v = \frac{ds}{dt}$, la loi fondamentale de Newton (1687), remaniée par Euler (1747) en équation différentielle, devient

$$\begin{aligned} \frac{ds}{dt} &= v \\ \frac{dv}{dt} &= -\frac{p}{\sqrt{1+p^2}}\ . \end{aligned} \tag{1.1}$$

Cela ne suffit pas ; à chaque instant nous devons connaître le p, pour notre exemple $p = y' = 4x^3 - 4x$, Ce qui nécessite la connaissance de x. Pour son calcul, nous rajoutons le x avec $\frac{dx}{dt} = \frac{dx}{ds} \cdot \frac{ds}{dt}$ à nos équations différentielles comme troisième équation ;

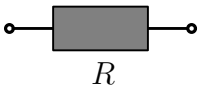
$$\frac{dx}{dt} = \frac{v}{\sqrt{1+p^2}} \qquad \text{avec} \quad p = 4x^3 - 4x\ . \tag{1.2}$$

Les trois équations (1.1) et (1.2) forment un *système d'équations différentielles* pour les trois fonctions $s(t), v(t), x(t)$; elles permettent de calculer numériquement ces fonctions ... ... seulement si on disposait de méthodes numériques pour leur calcul... !

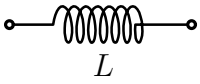
Deuxième exemple : le circuit de Van der Pol.

Balthasar Van der Pol était le scientifique en chef de la "Philips Glooilampenfabrieek" à Eindhoven, un des pionniers du développement des radios, puis téléviseurs en de tous les appareils éléctroniques qui nous entournent quotidiennement. Les ingrédiens qui font tout fonctionner sont les *circuits*, d'abord avec triodes, plus tard avec transisteurs, et finalement sur les chips .... Retraçons donc l'histoire de cette page fondamentale de la civilisation d'aujourd'hui.

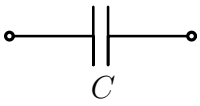
Petit excours en éléctronique: Soit I le courant (en Ampères) qui passe dans un conducteur, et soit U (en Volt) la tension entre deux noeuds. Alors on a les lois

Résistance : 

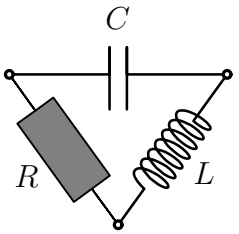
$U = I \cdot R$ (Ohm)

Inductivité : 

$U = L \cdot \dfrac{dI}{dt}$ (Faraday)

Condensateur : 

$I = C \cdot \dfrac{dU}{dt}$ (Capacité)



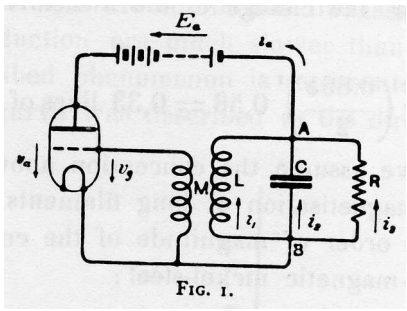


FIG. III.2: Circuit RCL (gauche) ; Fac-similé du circuit de Van der Pol (1926) (droite)

Si nous avons maintenant un circuit RCL (à gauche dans la Fig, .III.2), alors sont valables les *lois de Kirchhoff*, pour cet exemple $I_R = I_C = I_L$ et $U_R + U_C + U_L = 0$, i.e.,

$$I \cdot R + L \cdot \frac{dI}{dt} + \frac{1}{C} \cdot \int I \, dt = 0$$

ou, après différentiation,

$$L \cdot \frac{d^2I}{dt^2} + R \cdot \frac{dI}{dt} + \frac{1}{C} \cdot I = 0 \, .$$

Cette équation est de la forme (si nous prenons $LC = 1$)

$$y'' + \alpha y' + y = 0 \;\; \Rightarrow \left(y = e^{\lambda t}\right) \Rightarrow \;\; \lambda^2 + \alpha\lambda + 1 = 0, \quad y = e^{-\frac{\alpha}{2}x}(c_1 \cos \omega x + c_2 \sin \omega x),$$

une oscillation *amortie*.

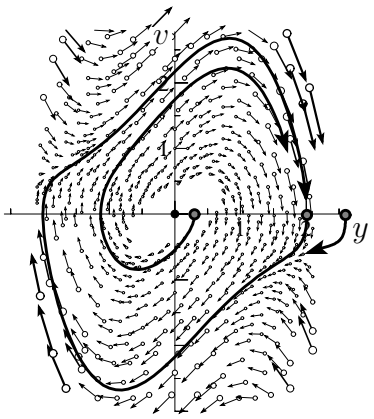
Idée. On rajoute une *triode* (voir Fig. III.2 à droite) et une batterie. Ainsi, la résistance R (ou α) va dépendre de y. Ceci a comme effet *d'augmenter l'énérgie pour y petit* et nous arrivons à

$$y'' + f(y) \cdot y' + y = 0 \qquad \begin{array}{l} f(y) < 0 \text{ si } y \text{ petit} \, ; \\ f(y) > 0 \text{ si } y \text{ grand.} \end{array} \tag{1.3}$$

Le modèle le plus simple est

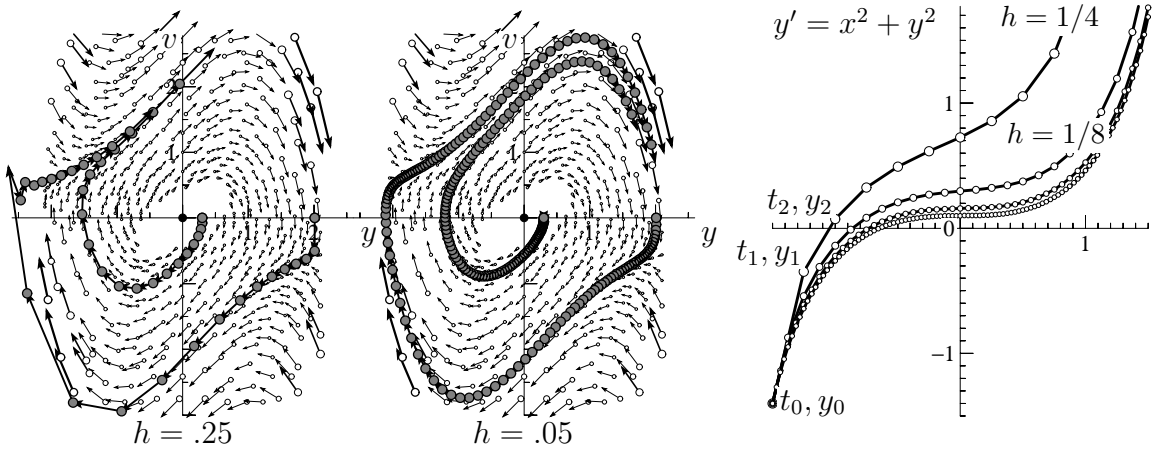
$$y'' + \epsilon(y^2 - 1)y' + y = 0 \qquad \text{ou} \qquad \begin{aligned} y' &= v, \\ v' &= \epsilon(1 - y^2)v - y \end{aligned} \tag{1.4}$$

où $\epsilon > 0$ est un paramètre.
Cette équation détermine, pour chaque point (y, v) donné, la vitesse que le "point mobile" (dans le language de Poincaré 1983) $(y(t), z(t))$ doit possèder. On appelle cela un *champs de vecteurs* (voir dessin). Une courbe qui satisfait cette vitesse à chaque instant, est *solution* du problème. Pas question de la trouver analytiquement!!!



Nous obsérvons un fait intéressant : cette équation possède un mouvement périodique qui s'appèlle (aussi dans le language de Poincaré) un *cycle limite*. Cela fait fonctionner nos appareils.!!!

III.2 Méthode d'Euler



Pour calculer une approximation de la solution de

$$y' = f(x, y), \quad y(x_0) = y_0 \tag{2.1}$$

sur l'intervalle $[x_0, \bar{x}]$, on procède comme suit: on subdivise $[x_0, \bar{x}]$ en sous-intervalles d'extrémités $x_0 < x_1 < \ldots < x_N = \bar{x}$, on dénote $h_n = x_{n+1} - x_n$ et on calcule l'approximation $y_n \approx y(x_n)$ par la formule (Euler 1768)

$$y_{n+1} = y_n + h_n f(x_n, y_n). \tag{2.2}$$

Une telle formule s'appelle "méthode à *un* pas", car le calcul de y_{n+1} utilise uniquement les valeurs h_n, x_n, y_n et non $h_{n-1}, x_{n-1}, y_{n-1}, \ldots$. Notre but est de démontrer la convergence de cette méthode si les $h_n \to 0$. La même preuve se laisse aussi adapter pour prouver la convergence des méthodes d'ordre supérieur.

Erreur locale.

Pour simplifier la notation nous considérons uniquement le premier pas ($n = 0$) dans (2.2) et nous notons $h_0 = h$) :

$$y_1 = y_0 + hf(x_0, y_0)\ . \tag{2.3}$$

Pour trouver l'erreur locale de cette approximation, nous utilisons la série de Taylor

$$y(x_0+h) = y_0 + hy_0' + \frac{h^2}{2!}y_0'' + \frac{h^3}{3!}y_0''' + \dots \tag{2.4}$$

ou, parfois, la série tronquée avec la dernière dérivée évaluée dans un point intermédiaire (formule de Lagrange [HW97], Thm. III.7.14). Les dérivées supérieures sont obtenues en dérivant successivement l'équation différentielle (2.1). Pour simplifier ces calculs, nous absorbons la variable x dans les y en augmentant la dimension par 1 et dérivons (voir (0.4) ci-dessus) $y'(x) = f(y(x))$. La dérivée en chaîne de cette formule donne

$$y''(x) = f'(y(x))y'(x) = f'(y(x))f(y(x)) \tag{2.5}$$

où $f'(y)$ est la matrice jabobienne de f. En comparant (2.2) et (2.4), nous voyons (la méthode d'Euler n'est rien d'autre que les deux premiers termes de la série de Taylor)

$$\|y(x_0+h) - y_1\| \le C \cdot h^2 \qquad \text{où} \qquad C = \max_U \|\frac{f'f}{2!}\| \,. \tag{2.6}$$

Similairement aux formules de quadrature, nous appelons cela *une méthode d'ordre 1*.

Erreur globale.

Contrairement aux intégrales, où l'erreur globale a été tout simplement la somme des erreurs locales, nous avons ici le phénomène "du papillon" : les erreurs locales peuvent augmenter si on les propage avec le temps (voir figure III.3).

Théorème 2.1 (Cauchy 1824) *Soit $y(x)$ la solution de $y' = f(x,y)$, $y(x_0) = y_0$ sur l'intervalle $[x_0, X]$.*
Supposons que

a) l'erreur locale satisfasse l'estimation (2.6) dans un voisinage U de la solution ;
b) la fonction $f(x,y)$ satisfasse une "condition de Lipschitz"

$$\|f(x,y) - f(x,z)\| \le L \cdot \|y - z\| \tag{2.7}$$

pour $h \le h_{\max}$ dans le même voisinage.

Alors, l'erreur globale admet pour $x_n \le X$ l'estimation

$$\|y(x_n) - y_n\| \le h \cdot \frac{C}{L} \cdot \left(e^{L(x_n - x_0)} - 1\right) \tag{2.8}$$

où $h = \max_i h_i$, sous la condition que h soit suffisamment petit. La convergence est donc assurée.

Démonstration. L'idée est d'étudier l'influence de l'erreur locale, commise au $i^{\text{ème}}$ pas, sur l'approximation y_n. Ensuite, on va additionner les erreurs accumulées.

Propagation de l'erreur. Soient $\{y_n\}$ et $\{z_n\}$ deux solutions numériques avec pour valeurs initiales y_0 et z_0, respectivement. En utilisant $y_{n+1} = y_n + h_n f(x_n, y_n)$ et $z_{n+1} = z_n + h_n f(x_n, z_n)$ et la condition de Lipschitz (2.7), leur différence peut être estimée comme

$$\|y_{n+1} - z_{n+1}\| \le \|y_n - z_n\| + h_n L\|y_n - z_n\| = (1 + h_n L)\|y_n - z_n\| \le e^{h_n L}\|y_n - z_n\|. \tag{2.9}$$

Récursivement, on obtient alors

$$\|y_n - z_n\| \le e^{h_{n-1}L} \cdot e^{h_{n-2}L} \cdot \ldots \cdot e^{h_i L}\|y_i - z_i\| = e^{L(x_n - x_i)}\|y_i - z_i\|.$$

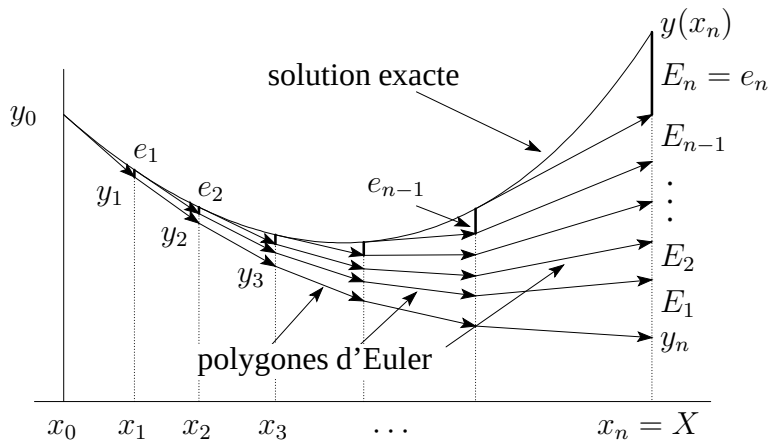


FIG. III.3: "Lady Windermere's Fan", Estimation de l'erreur globale

et l'erreur propagée E_i (voir la fig. III.3) satisfait

$$\|E_i\| \le e^{L(x_n-x_i)}\|e_i\| \le Ch_{i-1}^2 e^{L(x_n-x_i)}. \tag{2.10}$$

Accumulation des erreurs propagées. L'inégalité du triangle, ainsi que (2.10) nous donne (voir la fig. III.4 pour l'estimation de la somme)

$$\begin{aligned}\|y(x_n) - y_n\| &\le \sum_{i=1}^{n} \|E_i\| \le C\sum_{i=1}^{n} h_{i-1}^2 e^{L(x_n-x_i)} \\ &\le Ch\Big(h_0 e^{L(x_n-x_1)} + h_1 e^{L(x_n-x_2)} + \ldots + h_{n-2}e^{L(x_n-x_{n-1})} + h_{n-1}\Big) \\ &\le Ch\int_{x_0}^{x_n} e^{L(x_n-t)}\,dt = Ch\frac{1}{-L}e^{L(x_n-t)}\Big|_{x_0}^{x_n} = \frac{Ch}{L}\Big(e^{L(x_n-x_0)} - 1\Big).\end{aligned}$$

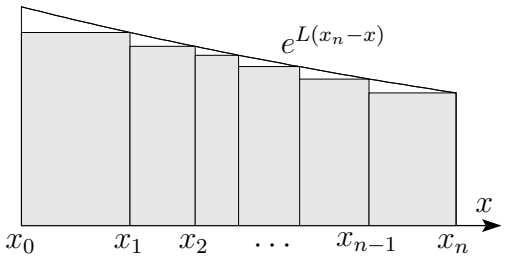


FIG. III.4: Estimation de la somme de Riemann

Il reste à justifier l'application de (2.7) dans (2.9), car l'estimation (2.7) n'est valable que dans un voisinage $U = \{(x,y)\,|\,\|y - y(x)\| \le b\}$ de la solution exacte. Si l'on suppose que h soit suffisamment petit, plus précisement si h est tel que

$$\frac{Ch}{L}\Big(e^{L(X-x_0)} - 1\Big) \le b,$$

on est sûr que toutes les solutions numériques de la fig. III.3 restent dans U. □

III.3 Méthodes "Predictor-Corrector"

Pour la dérivation d'autres méthodes numériques, intégrons (2.1) de x_0 à $x_0 + h$

$$y(x_0 + h) = y_0 + \int_{x_0}^{x_0+h} f(t, y(t))\,dt. \tag{3.1}$$

Idée "géniale" : remplacer l'intégrale de (3.1) par une formule de quadrature d'ordre plus élevé. Que nous donne, par exemple, la règle du trapèze :

$$y_1^* = y_0 + \frac{h}{2}\Big(f(x_0, y_0) + f(x_0 + h, y(x_0 + h))\Big) \,, \tag{3.2}$$

pour laquelle nous savons que

$$\|y(x_0 + h) - y_1^*\| \leq C \cdot h^3 \tag{3.3}$$

puisque cette formule de quadrature est d'ordre 2 (voir §I.2). Hélas !... La valeur de la solution exacte $y(x_0 + h)$, indispensable dans (3.2) à droite, n'est, bien sûr, pas connue : nous sommes en train de la calculer...

Idée. On "prédit" ("*Predictor*") la valeur manquante, qu'on appellera u_2 [1], par un pas d'Euler et on utilise (3.2) pour "corriger" la solution ("*Corrector*") :

$$\begin{aligned} u_2 &= y_0 + hf(x_0, y_0) \\ y_1 &= y_0 + \frac{h}{2}\Big(f(x_0, y_0) + f(x_0 + h, u_2)\Big) \end{aligned} \qquad \text{ou} \qquad \begin{aligned} u_2 &= y_0 + \frac{h}{2} f(x_0, y_0) \\ y_1 &= y_0 + hf(x_0 + \frac{h}{2}, u_2) \,. \end{aligned} \tag{3.4}$$

Règle du trapèze explicite. *Règle du point milieu expl.*

Ces deux méthodes, inventées par Runge en 1895, marquent le début d'un long développement des méthodes modernes.

Théorème 3.1 *Les deux méthodes de Runge sont de l'ordre 2, i.e., leur erreur locale satisfait*

$$\|y(x_0 + h) - y_1\| \leq C \cdot h^3 \,. \tag{3.5}$$

Preuve. On soustrait (3.2) et la deuxième ligne de (3.4) :

$$\|y_1^* - y_1\| = \frac{h}{2}\|f(x_0+h, y(x_0+h)) - f(x_0+h, u_2)\| \leq \frac{Lh}{2}\|y(x_0+h) - u_2\| \leq \frac{Lh}{2} \cdot C \cdot h^2$$

en utilisant (2.6). La preuve se termine par l'addition de cette inégalité à (3.3). Le clou de la preuve : l'erreur du predictor est multipliée par un h supplémentaire. Q.E.D.

La méthode de Heun d'ordre 3. Car

$$\int_0^1 x\Big(x - \frac{2}{3}\Big)\, dx = 0 \,,$$

la formule de quadrature (dite "de Radau")

$$\int_0^1 g(t)\, dt \approx \Big(\frac{1}{4}g(0) + \frac{3}{4}g(\,\frac{2}{3}\,)\Big)$$

est, par superconvergence (voir §I.3), d'ordre 3. Pour en faire une méthode pour équations différentielles du même ordre, nous devons "prédire" la valeur de $y(x_0 + \frac{2}{3}h)$ au moins à l'ordre 2. Faisons le avec la méthode du point milieu avec h remplacé par $\frac{2h}{3}$. Cela donne (Heun 1900) :

$$\begin{aligned} u_2 &= y_0 + \frac{h}{3} f(x_0, y_0) \\ u_3 &= y_0 + \frac{2h}{3} f(x_0 + \frac{h}{3}, u_2) \\ y_1 &= y_0 + h\Big(\frac{1}{4}f(x_0, y_0) + \frac{3}{4}f(x_0 + \frac{2h}{3}, u_3)\Big) \end{aligned} \tag{3.6}$$

[1] on comprendra plus tard pourquoi nous écrivons u_2 et non u_1.

Par une preuve similaire à la précédente, on voit que cette méthode *est d'ordre 3*.

Remarque historique. Le *premier* programme, qui a tourné sur le *premier* ordinateur (américain, i.e., l'ancêtre direct de tous nos ordinateurs), fut une équation différentielle résolue par la méthode de Heun.

Pour une interprétation géométrique de ces trois méthodes, voir figure III.5.

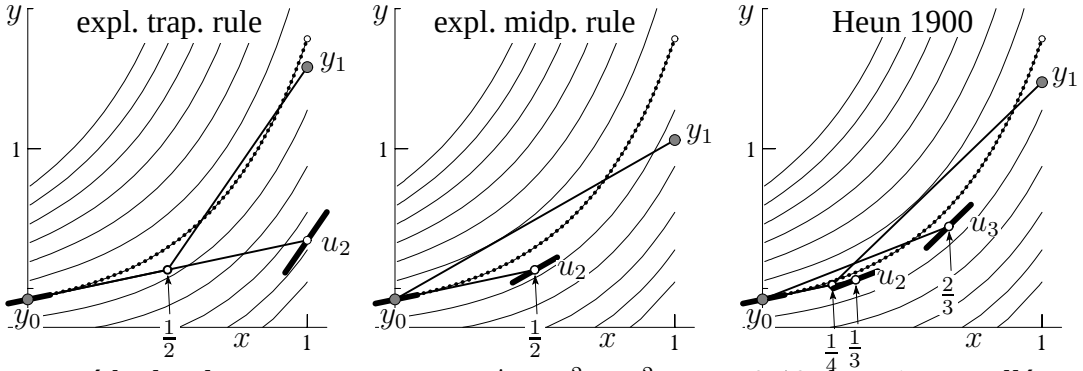


FIG. III.5: Méthodes de Runge-Kutta pour $y' = x^2 + y^2$, $y_0 = 0.46$, $h = 1$; pointillé: solution exacte.

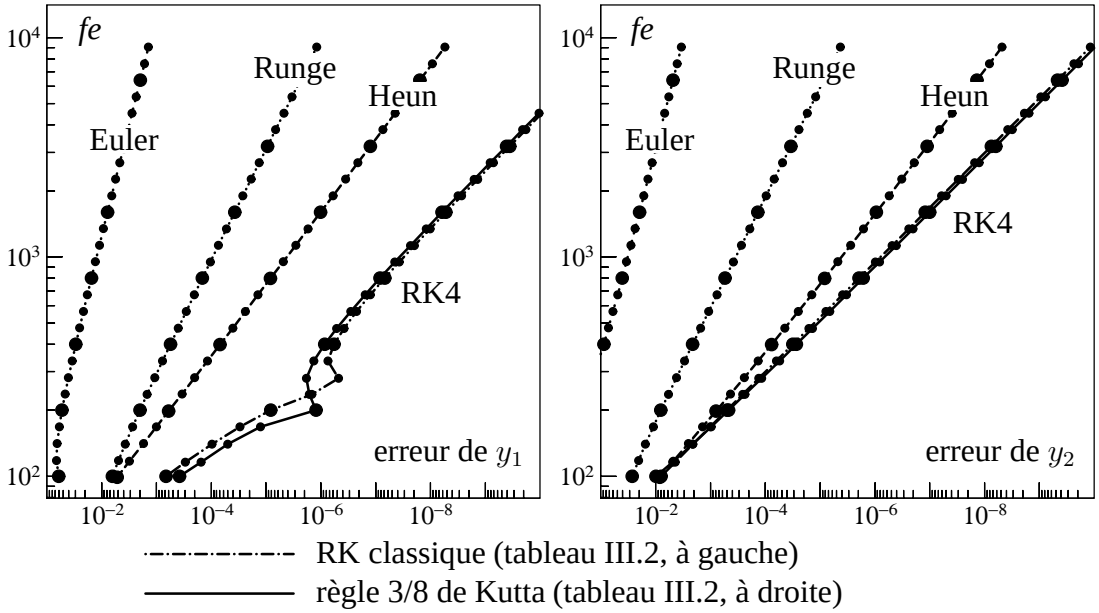


FIG. III.6: Erreur globale par rapport au travail numérique (chaque 4ème bille correspond à $h \mapsto \frac{h}{2}$).

Expérience numérique. Considérons les trois méthodes ci-dessus, ainsi que les deux méthodes Runge-Kutta d'ordre 4 du paragraphe suivant (Tableau III.2) ; comparons leur performance pour le problème de Van der Pol (voir §III.1)

$$\begin{aligned} y_1' &= y_2 & y_1(0) &= 2.00861986087484313650940188 \\ y_2' &= (1-y_1^2)y_2 - y_1 & y_2(0) &= 0 \end{aligned} \tag{3.7}$$

Les valeurs initiales ont été choisies pour rendre la solution périodique. Nous subdivisons l'intervalle de périodicité $[0, T]$, où $T = 6.6632868593231301896996820305$, en n parties équidistantes et appliquons n fois la méthode. Le travail (nombre total d'évaluations de f) est alors dessiné en fonction de l'erreur à la fin de l'intervalle (fig. III.6). Comme dans la fig. I.5 (intégration numérique), on peut constater que $\log_{10}(\text{fe})$ dépend linéairement de $-\log_{10}(err)$ et que cette droite est de pente $1/p$, où p est l'ordre de la méthode. Il est donc important d'utiliser des méthodes d'ordre élevé.

III.4 Méthodes de Runge-Kutta

La méthode pour équations différentielles fut, presque pendant un siècle, **la** *méthode de Runge-Kutta* d'ordre 4, trouvée par W. Kutta 1901 en généralisant les méthodes de Runge et de Heun. **Aucun** texte d'analyse numérique ne peut se passer de donner ces formules célèbres

$$\begin{aligned} u_1 &= y_0 \\ u_2 &= y_0 + \frac{h}{2} f(x_0, u_1) \\ u_3 &= y_0 + \frac{h}{2} f(x_0 + \frac{h}{2}, u_2) \\ u_4 &= y_0 + h f(x_0 + \frac{h}{2}, u_3) \\ y_1 &= y_0 + h\Big(\frac{1}{6} f(x_0, u_1) + \frac{2}{6} f(x_0 + \frac{h}{2}, u_2) + \frac{2}{6} f(x_0 + \frac{h}{2}, u_3) + \frac{1}{6} f(x_0 + h, u_4)\Big) \end{aligned} \tag{4.1}$$

pratiquement **aucun** texte n'explique au lecteur, comment ces formules ont été trouvées. Évidemment, dans une des meilleures universités d'Europe [2] et dans un des meilleurs cours de la faculté [3], on est obligé d'en faire exception . . . . : -)

Forme générale d'une méthode de Runge-Kutta. Dans l'esprit des formules de quadrature de Gauss, on introduit des coefficients arbitraires b_i, a_{ij} et $c_i = \sum_j a_{ij}$. Pour simplifier les formules et la preuve, nous rajoutons la variable x aux variables y avec $x' = 1$, comme nous avons déjà fait en (0.4) et en §III.2. Ainsi l'algorithme est

$$\begin{aligned} u_1 &= y_0 \\ u_2 &= y_0 + h a_{21} f(u_1) \\ u_3 &= y_0 + h\Big(a_{31} f(u_1) + a_{32} f(u_2)\Big) \\ u_4 &= y_0 + h\Big(a_{41} f(u_1) + a_{42} f(u_2) + a_{43} f(u_3)\Big) \\ \dots &\quad \dots \\ y_1 &= y_0 + h\Big(b_1 f(u_1) + b_2 f(u_2) + b_3 f(u_3) + b_4 f(u_4) + \dots\Big) \end{aligned} \tag{4.2}$$

On a l'habitude de représenter les coefficients à l'aide du schéma

c_i	a_{ij}
	b_i

Exemples. La méthode d'Euler, ainsi que des méthodes de Runge et de Heun sont données dans le tableau III.1. Deux méthodes de Kutta dans le tableau III.2.

TAB. III.1: Les premières méthodes de Runge-Kutta

0		0			0			0							
		1	1		1/2	1/2		1/3	1/3						
	1		1/2	1/2		0	1	2/3	0	2/3					
									1/4	0	3/4				

[2] paroles du Recteur ...
[3] paroles du Doyen ...

TAB. III.2: Méthodes de Kutta (1901)

0				
1/2	1/2			
1/2	0	1/2		
1	0	0	1	
	1/6	2/6	2/6	1/6

"La" méthode de Runge-Kutta

0				
1/3	1/3			
2/3	−1/3	1		
1	1	−1	1	
	1/8	3/8	3/8	1/8

règle 3/8

Conditions d'ordre pour méthodes RK

Problème. Trouver des conditions pour les coefficients b_i, a_{ij} et $c_i = \sum_j a_{ij}$ pour que (4.2) représente une méthode d'ordre p, i.e., s'assurer que l'erreur locale est de $C \cdot h^{p+1}$.

L'*idée* est simple : calculer les dérivées successives de y_1, qui est définie par cet amalgame de formules, par rapport à h et s'assurer que ces dérivées sont les mêmes que celles de la solution exacte.

Seulement, l'*exécution* de ce plan est loin d'être triviale, si on ne fait pas attention. Pour convaincre le lecteur, nous représentons en figure III.7 deux pages (de huit en total) du calcul de la sixième dérivée pour une méthode avec $s = 8$ par Huťa (1956).

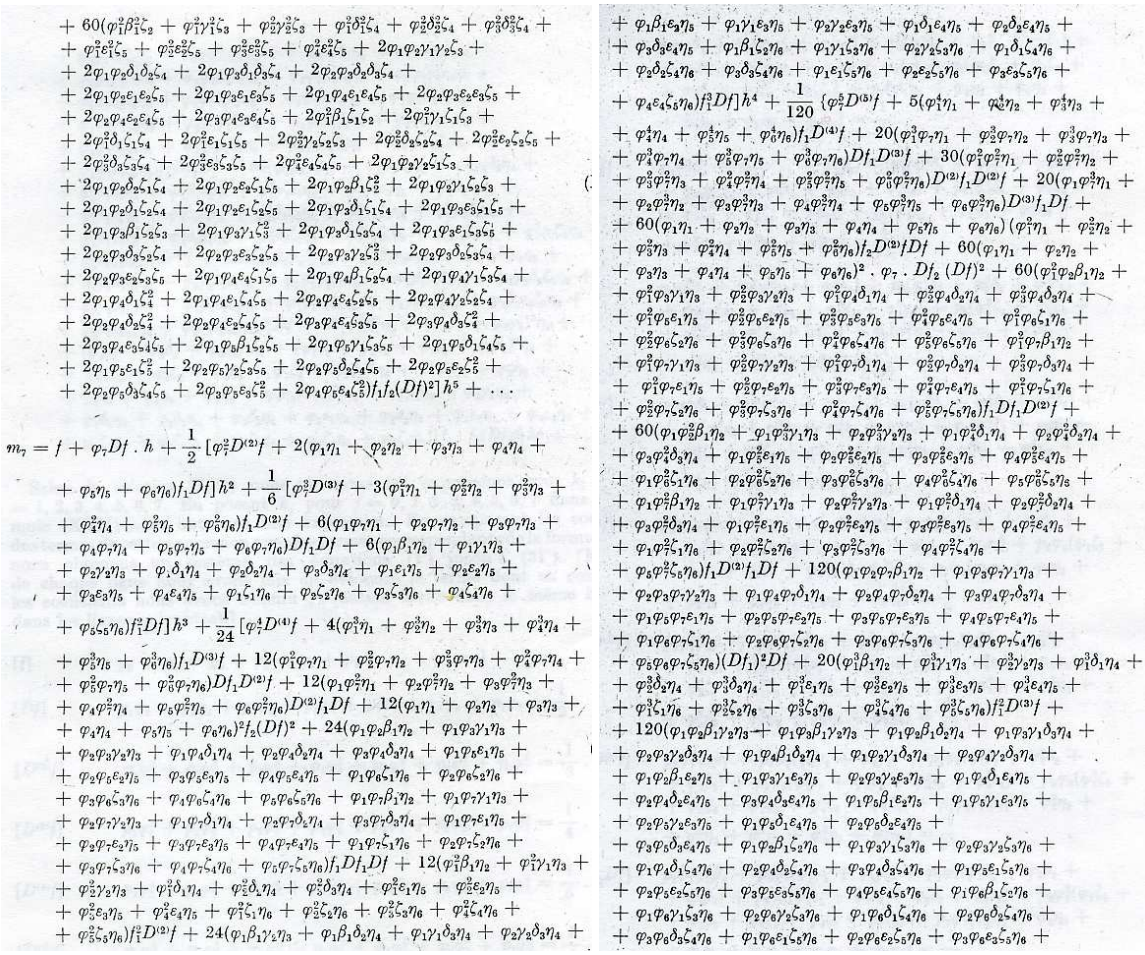


FIG. III.7: Sixième dérivée de y_1 d'une méthode avec $s = 8$ calculée par Huťa (1956) (extrait)

Pour l'*élégance* il fallait attendre les travaux de Merson (1957) et surtout de John Butcher (travaux de 1963–1972), M. Crouzeix, E. Hairer et Chr. Lubich. Nous écrivons la méthode sous la forme

$$u_\sigma = y_0 + h\sum_i a_{\sigma i} f(u_i)\ , \qquad y_1 = y_0 + h\sum_i b_i f(u_i)\ . \tag{4.3}$$

Nous calculons les dérivées de u_σ ; pour celles de y_1 il suffit ensuite de remplacer le $a_{\sigma i}$ par b_i.

La fonction (de h) u_σ est de la forme $h\cdot g(h)$, dont les dérivées sont, par Leibniz,

$$u'_\sigma = 1\cdot g(h) + hg'(h), \qquad u''_\sigma = 2\cdot g'(h) + hg''(h), \qquad u'''_\sigma = 3\cdot g''(h) + hg'''(h), \quad \ldots \tag{4.4}$$

Pour $h=0$, seulement les premiers termes restent non nuls. Ainsi, par (4.3),

$$u'_\sigma = 1\cdot\sum_i a_{\sigma i} f \tag{4.5}$$

où u'_σ est évalué en $h=0$ et f en y_0. La deuxième dérivée devient par (4.4) et (4.3)

$$u''_\sigma = 2\cdot\sum_i a_{\sigma i}\ (f(u_i))' = 2\cdot\sum_i a_{\sigma i}\ f'\cdot u'_i\ . \tag{4.6}$$

Dans cette formule, nous devons insérer u'_i de (4.5). Pour éviter une confusion des indices, nous faisons le shift $i\mapsto j$, $\sigma\mapsto i$. Cela donne

$$u''_\sigma = 2\cdot 1\cdot\sum_{i,j} a_{\sigma i}a_{ij}\ f'\cdot f \qquad \text{et} \quad y''_1 = 2\cdot 1\cdot\sum_{i,j} b_i a_{ij}\ f'\cdot f\ . \tag{4.7}$$

Nous avons vu en §III.2, formule (2.5), que la deuxième dérivée de la solution exacte est $y''=f'\cdot f$, précisément la formule pour la deuxième dérivée de la solution numérique **sans** les facteurs $2\cdot 1$ et $\sum_{i,j} b_i a_{ij}$. Cela donne déjà notre premier théorème :

Théorème 4.1 *La méthode (4.2) est de l'ordre 2 ssi*

$$\sum_i b_i = 1 \qquad \textit{et} \qquad \sum_{i,j} b_i a_{ij} = \frac{1}{2}\,.$$

À cause de $c_i=\sum_j a_{ij}$, la deuxième condition devient $\sum_i b_i c_i = 1/2$. On peut ainsi vérifier que les deux méthodes de Runge sont effectivement d'ordre 2.

La troisième dérivée. La troisième dérivée devient par (4.4) et (4.3)

$$u'''_\sigma = 3\cdot\sum_i a_{\sigma i}\ (f(u_i))'' = 3\cdot\sum_i a_{\sigma i}\ (f'\cdot u'_i)' = 3\cdot\sum_i a_{\sigma i}\ \left(f''(u'_i,u'_i) + f'\cdot u''_i\right) \tag{4.8}$$

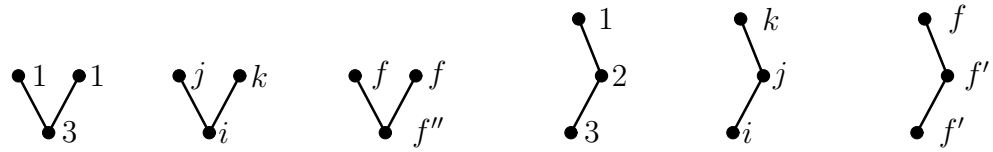
Dans cette formule, nous devons insérer u'_i de (4.5) et u''_i de (4.7) avec, de nouveau, des shifts d'indices appropriés. Cela donne

$$u'''_\sigma = 3\cdot 1\cdot 1\cdot\sum_{i,j,k} a_{\sigma i}\,a_{ij}\,a_{ik}\cdot\ f''(f,f) + 3\cdot 2\cdot 1\cdot\sum_{i,j,k} a_{\sigma i}\,a_{ij}\,a_{jk}\cdot f'(f'(f)) \tag{4.9}$$

et la formule analogue pour y'''_1. Avant que ces calculs ne tournent au cauchemar, nous observons une belle analogie de ces formules avec les *arbres orientés*

de manière suivante :

$$y_1''' = \underset{\uparrow}{3\cdot 1\cdot 1}\cdot \sum_{i,j,k} b_i\, a_{ij}\, a_{ik} \underset{\uparrow}{\cdot f''(f,f)} + \underset{\uparrow}{3\cdot 2\cdot 1}\cdot \sum_{i,j,k} b_i\, a_{ij}\, a_{jk} \underset{\uparrow}{\cdot f'(f'(f))}$$



Nous appelons *ordre d'un arbre* le nombre de ses noeud. La troisième dérivée est représentée par tous les arbres d'ordre 3. On y voit apparaître trois facteurs :

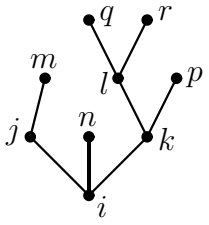
- Un produit de nombres entiers (les ordres de tous les sous-arbres) ;
- une somme de produits a_{jk}, où les indices jk sont connectes dans la même façon que les noeuds de l'arbre ; le premier facteur étant b_i ;
- les dérivées $f^{(q)}$ interconnectées comme applications multilinéaires comme les noeuds correspondants de l'arbre.

Le troisième facteur (qui pour les ordres plus élévés contient encore un entier supplémentaire) apparaît aussi dans la solution exacte. Nous avons donc :

Théorème 4.2 *La méthode est d'ordre p ssi le produit du premier et deuxième facteur est 1 pour chaque arbre dont le nombre de noeuds est $\leq p$.*

Exemple 4.3 *Pour l'arbre suivant d'ordre 9 nous avons*

$$\sum_{i,j,k,l,m,n,p,q,r} b_i a_{ij} a_{jm} a_{in} a_{ik} a_{kl} a_{lq} a_{lr} a_{kp} = \frac{1}{9\cdot 2\cdot 5\cdot 3}$$



ou bien, par $\sum_j a_{ij} = c_i$,

$$\sum_{i,j,k,l} b_i c_i a_{ij} c_j a_{ik} c_k a_{kl} c_l^2 = \frac{1}{270}.$$

III.5 Construction de méthodes d'ordre 4

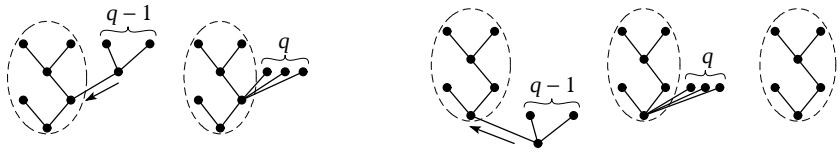
Le but est de déterminer les coefficients a_{ij}, b_j et $c_i = \sum_j a_{ij}$ afin que l'erreur locale soit $C\cdot h^5$. Il existent huit arbres d'ordre 4 et nous avons :

Théorème 5.1 (conditions d'ordre) *La méthode de Runge-Kutta (4.2) a l'ordre 4 si les coefficients satisfont*

$$\begin{aligned}
\textstyle\sum_i b_i &= 1 && (= b_1+b_2+b_3+b_4) && (5.1a)\\
\textstyle\sum_i b_i c_i &= 1/2 && (= b_2c_2+b_3c_3+b_4c_4) && (5.1b)\\
\textstyle\sum_i b_i c_i^2 &= 1/3 && (= b_2c_2^2+b_3c_3^2+b_4c_4^2) && (5.1c)\\
\textstyle\sum_{i,j} b_i a_{ij} c_j &= 1/6 && (= b_3a_{32}c_2 + b_4(a_{42}c_2+a_{43}c_3)) && (5.1d)\\
\textstyle\sum_i b_i c_i^3 &= 1/4 && (= b_2c_2^3+b_3c_3^3+b_4c_4^3) && (5.1e)\\
\textstyle\sum_{i,j} b_i c_i a_{ij} c_j &= 1/8 && (= b_3c_3a_{32}c_2 + b_4c_4(a_{42}c_2+a_{43}c_3)) && (5.1f)\\
\textstyle\sum_{i,j} b_i a_{ij} c_j^2 &= 1/12 && (= b_3a_{32}c_2^2 + b_4(a_{42}c_2^2+a_{43}c_3^2)) && (5.1g)\\
\textstyle\sum_{i,j,k} b_i a_{ij} a_{jk} c_k &= 1/24 && (= b_4a_{43}a_{32}c_2). && (5.1h)
\end{aligned}$$

(entre parenthèses on a explicité les expressions pour $s=4$). □

Résolution du système (5.1) pour $s = 4$. Kutta (dans sa thèse 1901) donne les solutions de ces 8 équations nonlinéaires à 10 inconnues sans le moindre commentaire. Mais si plus tard on désirait passer à l'ordre 5 (17 équations) ou 6 (37 équations) ou 8 (200 équations) ou même 10 (1206 équations), il vaut mieux s'équiper de quelques idées et de regarder un peu plus attentivement la structure de ces équations. La clef sont les conditions suivantes, découverte par John Butcher 1963:



$$\sum_{j=1}^{s} a_{ij} c_j^{q-1} = \frac{c_i^q}{q} \qquad \sum_{i=1}^{s} b_i c_i^{q-1} a_{ij} = \frac{b_j}{q}\,(1 - c_j^q)$$

Ces conditions permettent de réduire la condition pour l'arbre de gauche à celui des arbres de droite (essayez un exemple). Nous utilisons ici la deuxième condition pour $q = 1$:

$$\sum_{i=j+1}^{s} b_i a_{ij} = b_j(1 - c_j). \tag{5.2}$$

c.-à-d.

$$b_2(1 - c_2) = b_3 a_{32} + b_4 a_{42} \tag{5.3a}$$
$$b_3(1 - c_3) = b_4 a_{43} \tag{5.3b}$$
$$b_4(1 - c_4) = 0. \tag{5.3c}$$

Un calcul direct montre que les trois conditions (5.1d,g,h) peuvent être remplacées par (5.3a,b,c) sans changer de solutions. Grand avantage : si les b_i et les c_i sont choisis, ces équations sont **linéaires**.

Algorithme. *Poser $c_1 = 0$, $c_4 = 1$; c_2 et c_3 sont des paramètres libres; calculer b_1, b_2, b_3, b_4 tels que la formule de quadrature soit d'ordre 4 (conditions (5.1a,b,c,e)); calculer a_{43} de (5.3b), a_{42} et a_{32} du système linéaire (5.1f)–(5.3a); finalement calculer a_{21}, a_{31}, a_{41} de c_i pour $i = 2, 3, 4$.*

Parmi cette classe de méthodes d'ordre 4, les plus celèbres sont données dans le tableau III.2. La "RK solution" de la fig. III.6 a été obtenue par la méthode de gauche. Elle est basée sur la formule de Simpson.

La lutte pour des méthodes d'ordre supérieur.
La table III.3 résume les résultats principaux de cette lutte, qui s'étend sur presqu'un siècle. Cette table est reprise d'un article sur l'histoire des RK par Butcher–Wanner (1996).

III.6 Un programme à pas variables

Pour résoudre un problème réaliste, un calcul à pas constants est en général inefficace. Mais comment choisir la division? L'idée est de choisir les pas afin que l'erreur locale soit partout environ égale à *Tol* (fourni par l'utilisateur). A cette fin, il faut connaître une estimation de l'erreur locale. Inspiré par le programme TEGRAL pour l'intégration numérique (voir I.6), nous construisons une deuxième méthode de Runge-Kutta avec $\widehat{y}_1$ comme approximation numérique, et nous utilisons la différence $\widehat{y}_1 - y_1$ comme estimation de l'erreur locale du moins bon résultat.

TAB. III.3: Successive derivations of high order Runge-Kutta methods

p	s	Auteur	Année		Nr. d'équ.
2	2	Coriolis	1837	(Trapezoidal rule method)	2
2	2	Runge	1895	(rediscovery of Trap. rule method)	2
2	2	Runge	1895	(Midpoint rule method)	2
3	4	Runge	1895		4
3	3	Heun	1900		4
4	8	Heun	1900		8
4	4	Kutta	1901	(Table III.2)	8
5	6	Kutta	1901		17
5	6	Nyström	1925	(correction to a method of Kutta)	17
6	8	Huťa	1956		37
6	7	Butcher	1964		37
7	9	Butcher		(known since approximately 1968)	85
8	11	Curtis	1970		200
8	11	Cooper–Verner	1972	(ann. 1969 in Verner's thesis)	200
10	18	Curtis	1975		1205
10	17	Hairer	1978		1205

Méthode emboîtée. Soit donnée une méthode d'ordre p à s étages (coefficients c_i, a_{ij}, b_j). On cherche une approximation $\widehat{y}_1$ d'ordre $\widehat{p} < p$ qui utilise les mêmes évaluations de f, c.-à-d.,

$$\widehat{y}_1 = y_0 + h(\widehat{b}_1 k_1 + \ldots + \widehat{b}_s k_s) \tag{6.1}$$

où les k_i sont donnés par la méthode (**??**). Pour avoir plus de liberté, on ajoute souvent un terme contenant $f(x_1, y_1)$ à la formule (il faut en tous cas calculer $f(x_1, y_1)$ pour le pas suivant) et on cherche $\widehat{y}_1$ de la forme

$$\widehat{y}_1 = y_0 + h\Big(\widehat{b}_1 k_1 + \ldots + \widehat{b}_s k_s + \widehat{b}_{s+1} f(x_1, y_1)\Big). \tag{6.2}$$

Exemple. Prenons une méthode d'ordre $p = 4$ à $s = 4$ étages et cherchons une méthode emboîtée d'ordre $\widehat{p} = 3$ qui soit de la forme (6.2). Les conditions d'ordre sont obtenues par le théorème du paragraphe III.5 (on augmente s de 1 et on ajoute un $(s+1)^{\text{ème}}$ étage avec pour coefficients $a_{s+1,i} = b_i$ pour $i = 1, \ldots, s$) :

$$\widehat{b}_1 + \widehat{b}_2 + \widehat{b}_3 + \widehat{b}_4 + \widehat{b}_5 = 1 \tag{6.3a}$$
$$\widehat{b}_2 c_2 + \widehat{b}_3 c_3 + \widehat{b}_4 + \widehat{b}_5 = 1/2 \tag{6.3b}$$
$$\widehat{b}_2 c_2^2 + \widehat{b}_3 c_3^2 + \widehat{b}_4 + \widehat{b}_5 = 1/3 \tag{6.3c}$$
$$\widehat{b}_3 a_{32} c_2 + \widehat{b}_4 (a_{42} c_2 + a_{43} c_3) + \widehat{b}_5/2 = 1/6. \tag{6.3d}$$

Ceci représente un système linéaire de 4 équations pour 5 inconnues. On peut arbitrairement choisir $\widehat{b}_5$ et résoudre le système pour les autres variables. Pour le choix $\widehat{b}_5 = 1/6$, on obtient ainsi :

$$\begin{aligned} &\widehat{b}_1 = 2b_1 - 1/6, && \widehat{b}_2 = 2(1-c_2)b_2, \\ &\widehat{b}_3 = 2(1-c_3)b_3, && \widehat{b}_4 = 0, \qquad \widehat{b}_5 = 1/6. \end{aligned} \tag{6.4}$$

Calcul du h "optimal". Si l'on applique la méthode avec une certaine valeur h, l'estimation de l'erreur satisfait ($\widehat{p} < p$)

$$y_1 - \widehat{y}_1 = (y_1 - y(x_0+h)) + (y(x_0+h) - \widehat{y}_1) = \mathcal{O}(h^{p+1}) + \mathcal{O}(h^{\widehat{p}+1}) \approx C \cdot h^{\widehat{p}+1}. \tag{6.5}$$

Le h "optimal", noté par h_{opt}, est celui où cette estimation est proche de *Tol*:

$$\mathit{Tol} \approx C \cdot h_{\text{opt}}^{\widehat{p}+1}. \tag{6.6}$$

En éliminant C de (6.5) et de (6.6), on obtient

$$h_{\text{opt}} = 0.9 \cdot h \cdot \sqrt[\widehat{p}+1]{\frac{\mathit{Tol}}{\|y_1 - \widehat{y}_1\|}} \tag{6.7}$$

(le facteur 0.9 est ajouté pour rendre le programme plus sûr).

Algorithme pour la sélection automatique du pas. Au début, l'utilisateur fournit un sous-programme qui calcule la valeur de $f(x,y)$, les valeurs initiales x_0, y_0 et un premier choix de h.

A) Avec h, calculer y_1, $\mathit{err} = \|y_1 - \widehat{y}_1\|$ et h_{opt} de (6.7).

B) **If** $\mathit{err} \le \mathit{Tol}$ (le pas est accepté) **then**
$x_0 := x_0 + h, \quad y_0 := y_1, \quad h := \min(h_{\text{opt}}, x_{\text{end}} - x_0)$
else (le pas est rejeté)
$h := h_{\text{opt}}$
end if

C) Si $x_0 = x_{\text{end}}$ on a terminé, sinon on recommence en (A) et on calcule le pas suivant.

Remarques. Il est recommandé de remplacer (6.7) par

$$h_{\text{opt}} = h \cdot \min\Big(5, \max(0.2, 0.9(\mathit{Tol}/\mathit{err})^{1/(\widehat{p}+1)})\Big).$$

Pour la norme dans (6.7) on utilise en général

$$\|y_1 - \widehat{y}_1\| = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\Big(\frac{y_{i1} - \widehat{y}_{i1}}{\mathit{sc}_i}\Big)^2} \qquad \text{où} \qquad \mathit{sc}_i = 1 + \max(|y_{i0}|, |y_{i1}|) \tag{6.8}$$

($y_{i0}, y_{i1}, \widehat{y}_{i1}$ est la $i^{\text{ème}}$ composante de $y_0, y_1, \widehat{y}_1$, respectivement). Ceci représente un mélange entre erreur relative et erreur absolue.

Exemple numérique. Cet algorithme a été programmé (en utilisant la "règle $3/8$" et la méthode emboîtée (6.4)) et il a été appliqué au problème (une réaction chimique, le "Brusselator")

$$\begin{aligned} y_1' &= 1 + y_1^2 y_2 - 4y_1 \qquad & y_1(0) &= 1.5 \\ y_2' &= 3y_1 - y_1^2 y_2 & y_2(0) &= 3 \end{aligned} \tag{6.9}$$

sur l'intervalle $[0, 20]$. Les résultats obtenus avec $\mathit{Tol} = 10^{-4}$ sont présentés dans la fig. III.8:

i) en haut, les deux composantes de la solution avec tous les pas acceptés;

ii) au milieu les pas; les pas acceptés étant reliés, les pas rejetés étant indiqués par $\times$;

iii) les dessin du bas montre l'estimation de l'erreur locale *err*, ainsi que les valeurs exactes de l'erreur locale et de l'erreur globale.

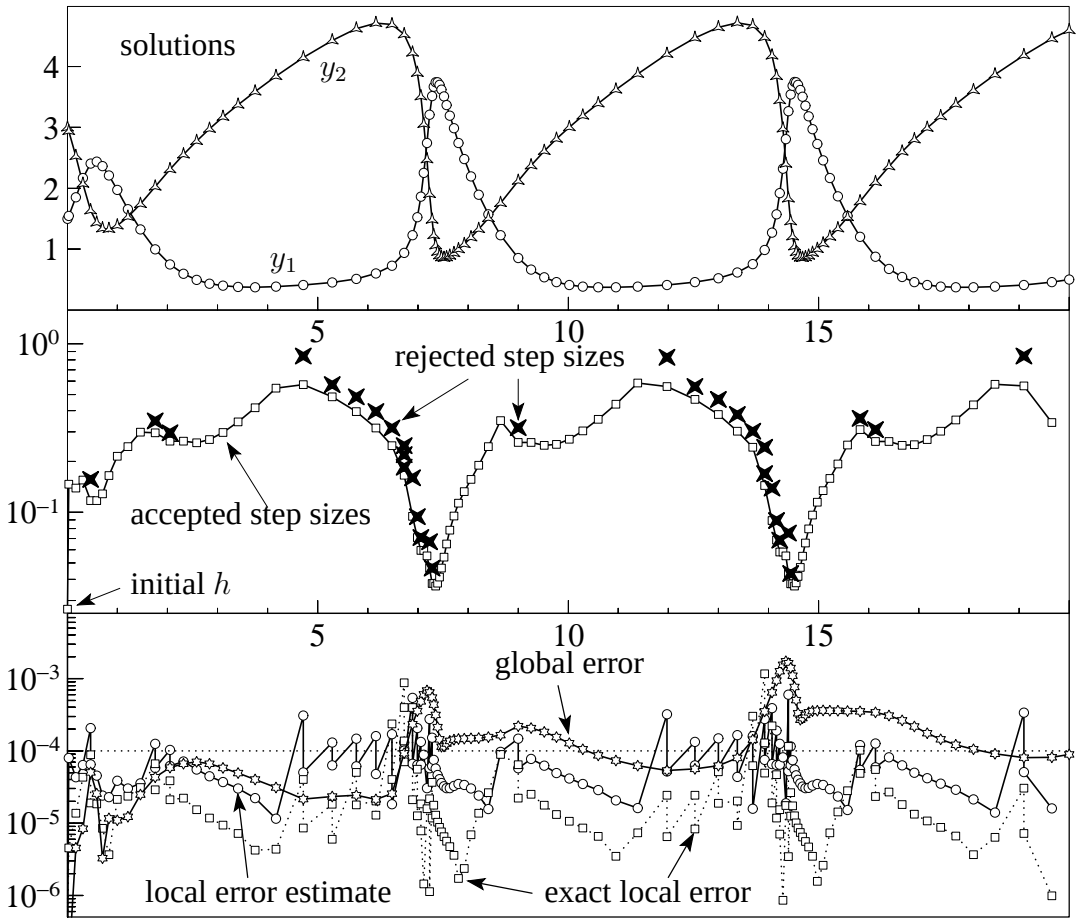


FIG. III.8: Sélection du pas, *Tol* $= 10^{-4}$, 96 pas acceptés + 32 pas rejetés

Les méthodes de Dormand et Prince.

Les méthodes actuellement les plus utilisées et les plus à conseiller (de loin) pour des calculs de type général sont celles de Dormand et Prince. La méthode DOPRI5, d'ordre 5 et d'ordre emboîtée 4, se distingue d'un choix particulièrement astucieux des paramètres libres (voir [HNW93], p. 178). On ne va pas apprendre les coefficients par cœur, mais télécharger le code depuis

`http://www.unige.ch/math/folks/hairer/software.html`.

Une deuxième méthode DOPRI8 (1989), d'ordre 8(6), a reçu sa dernière amélioration par E. Hairer (1993) et est devenue le code DOP853 (voir [HNW93], p. 181-185).