École Centrale Paris

Option Mathématiques appliquées

CSTD A
Méthodes numériques parallèles pour les équations différentielles ordinaires

Philippe Chartier [1]

2 octobre 1997

1. INRIA-Rennes, Campus de Beaulieu, 35042 Rennes Cedex, email : Philippe.Chartier@irisa.fr, tél : 02 99 84 74 00

Table des matières

Chapitre 1

Introduction

1.1 Problèmes de valeur initiale pour les systèmes différentiels

Dans tout ce cours, nous considérons le système d'équations différentielles ordinaires suivant

$$\left\{ \begin{array}{l} y'(x) = f(x, y(x)), \quad x \in [x_0, X] \\ y(x_0) = y_0 \in \mathbb{R}^m \end{array} \right. , \tag{1.1}$$

où f est une fonction de $[x_0, X] \times \mathbb{R}^m$ dans $\mathbb{R}^m$. On supposera toujours vérifiées les hypothèses nécessaires à l'existence et à l'unicité d'une solution continuement différentiable $y(x)$ sur tout l'intervalle d'intégration $[x_0, X]$. Ces conditions sont détaillées dans de nombreux ouvrages (voir par exemple [1], [6], [11] ou [13]).

1.2 Principales techniques de parallélisation

Le caractère «pas à pas» des méthodes de résolution numérique en général, -et des méthodes de Runge-Kutta en particulier-, est un obstacle intrinsèque à toute tentative de parallélisation. Il reflète en effet la nécessaire propagation de $y(x_0) = y_0$ à travers l'intervalle de gauche à droite, sans qu'il semble possible d'anticiper. La connaissance de y en un point détermine sa valeur en un point plus à droite. Au cours des 15 dernières années, diverses approches ont néanmoins été proposées. C.W. Gear (voir [12]) les a regroupées en trois grandes classes :

- parallélisme «à travers les équations»,
- parallélisme «à travers les pas»,
- parallélisme «à travers la méthode».

Chaque catégorie peut être envisagée indépendemment des deux autres. La première technique est simplissime et donc souvent fort efficace. Elle consiste à fractionner les évaluations de f suivant ses composantes. Si f a par exemple 10000 composantes et que l'on dispose de 100 processeurs, chaque processeur peut calculer 100 composantes. La seconde technique dérive du principe de l'itération de Picard. La solution de (1.1) est affinée itérativement à partir d'une solution initiale approchée. Son intérêt majeur réside dans la possibilité de découpler le système (1.1) en plusieurs sous-systèmes indépendants. Ses performances sont étonnantes dans le domaine de la simulation de composants électroniques, mais souvent désespérantes en dehors. Elle sera néanmoins abordée dans le dernier chapître. La troisième technique consiste à concevoir de nouvelles méthodes ou à en adapter d'autres afin de découpler certains calculs. Ainsi, une méthode de Runge-Kutta possédant une matrice A diagonale offre la possibilité de découpler le système $s \times m$ à résoudre à chaque pas en s sous-systèmes de dimension m. Il est malheureusement facile de voir qu'une telle méthode ne peut être d'ordre supérieur à 2. Mais d'autres idées de ce type ont connu plus de succès et seront développées dans les chapîtres suivants.

Chapitre 2

Méthodes de Runge-Kutta

2.1 Premiers exemples

La méthode la plus simple que l'on puisse construire (et historiquement la première née) est la méthode dite d'Euler :

$$\begin{aligned} y(x_0+h) \approx y_1 &= y_0 + hf(x_0, y_0) && \text{pour Euler explicite,} \\ &= y_0 + hf(x_0+h, y_1) && \text{pour Euler implicite.} \end{aligned}$$

Elle consiste à approcher la solution $y(x)$ de (1.1) sur le segment $[x_0, x_0+h]$ par une droite de

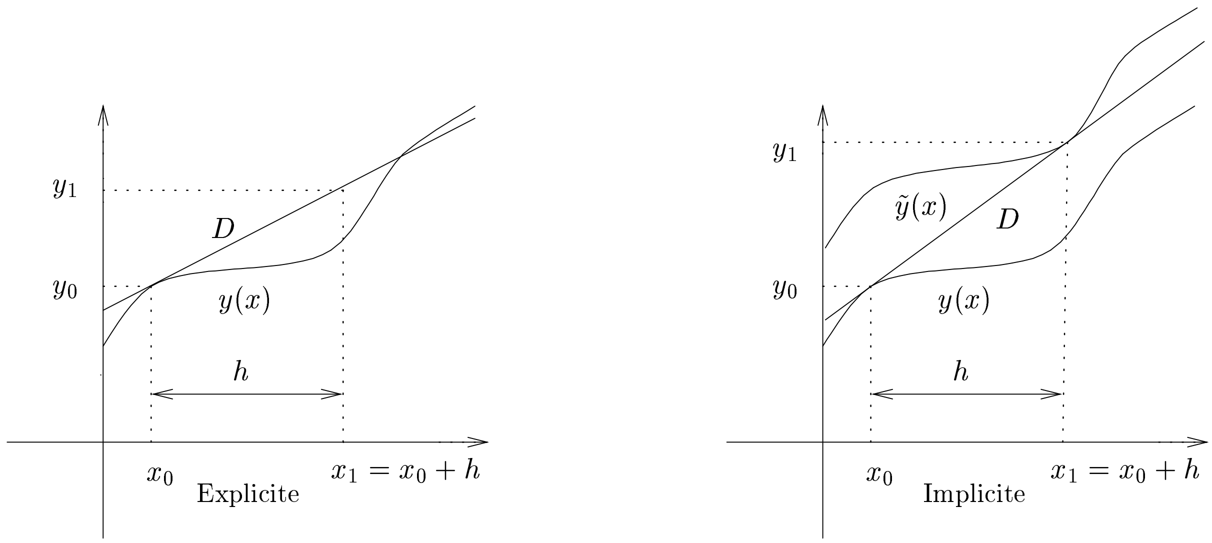


FIG. 2.1: *Méthodes d'Euler*

pente $f(x_0, y_0)$ ou $f(x_0+h, y_1)$: notons que la méthode d'Euler implicite nécessite à chaque pas la résolution d'un système *implicite* généralement non-linéaire, et est donc nettement plus coûteuse que son homologue explicite. Si la méthode d'Euler est extrêmement simple, elle est aussi peu précise. Intuitivement, il semble par exemple plus judicieux de calculer la pente de la droite D (voir Fig. 2.1) non pas en une extrémité du segment $[x_0, x_0+h]$ mais en son milieu. Cela conduit à prendre pour valeur approchée de $y(x_0+h)$,

$$y_1 = y_0 + hf\left(x_0 + \frac{h}{2}, y(x_0+h/2)\right).$$

La valeur de $y(x_0 + h/2)$ étant elle-même inconnue, on l'approche par

$$y(x_0 + h/2) \approx y_0 + \frac{h}{2} f(x_0, y_0).$$

Cette nouvelle méthode, due a Runge (1895), procède en deux étapes :

$$\begin{aligned} Y &= y_0 + \frac{h}{2} f(x_0, y_0), \\ y_1 &= y_0 + hf\left(x_0 + \frac{h}{2}, Y\right). \end{aligned}$$

On peut se convaincre de sa supériorité sur la méthode d'Euler en comparant les développements en série de Taylor de y_1 et de $y(x_0 + h)$ pour les deux méthodes. Pour Euler explicite, on a ainsi

$$\begin{aligned} y(x_0 + h) - y_1^{Euler} &= y(x_0) + hy'(x_0) + \frac{h^2}{2} y''(x_0) - y_0 - hf(x_0, y_0) + \mathcal{O}(h^3) \\ &= \frac{h^2}{2} y''(x_0) + \mathcal{O}(h^3). \end{aligned}$$

Il existe donc une constante C^{Euler} telle que $\|y(x_0 + h) - y_1^{Euler}\| \leq C^{Euler} h^2$. Pour la méthode de Runge, il vient

$$\begin{aligned} y(x_0 + h) - y_1^{Runge} &= y(x_0) + hy'(x_0) + \frac{h^2}{2} y''(x_0) + \frac{h^3}{6} y'''(x_0) - y_0 - hf\left(x_0 + \frac{h}{2}, Y\right) + \mathcal{O}(h^4) \\ &= (y(x_0) - y_0) + h\left(y'(x_0) - f(x_0, y_0)\right) + \frac{h^2}{2}\left(y''(x_0) - (f_x + f_y f)(x_0, y_0)\right) \\ &\quad + \frac{h^3}{6}\left(y'''(x_0) - \frac{6}{8}\left(f_{xx} + 2f_{xy} f + f_{yy}(f, f)\right)(x_0, y_0)\right) + \mathcal{O}(h^4). \end{aligned}$$

Il existe donc une constante C^{Runge} telle que

$$\|y(x_0 + h) - y_1^{Runge}\| \leq C^{Runge} h^3. \tag{2.1}$$

Lorsque l'une et l'autre de ces méthodes sont répétées jusqu'à l'extrémité droite du segment d'intégration, une puissance du pas h est perdue dans l'estimation de l'erreur dite *globale*, en raison de l'accumulation des erreurs dites *locales*. Ainsi, la méthode de Runge est d'ordre 2 (erreur en h^2), alors que la méthode d'Euler n'est que d'ordre 1 (erreur en h^1) : elle est plus précise dès que h est suffisamment petit.

2.2 Formulation générale

Une méthode de Runge-Kutta à s étapes («stages» en anglais) est définie par la donnée de 2 vecteurs b et c à s composantes et d'une matrice carrée A de taille $s \times s$.

Définition 1 *Soit s un entier non nul, $c = (c_1, \dots, c_s)^T$, $b = (b_1, \dots, b_s)^T$ deux vecteurs de $\mathbb{R}^s$, et $A = (a_{ij})_{1 \leq i,j \leq s}$ une matrice de $\mathbb{R}^{s \times s}$. La méthode*

$$Y_i = y_0 + h \sum_{j=1}^{s} a_{ij} f(x_0 + c_j h, Y_j), \quad i = 1, \dots, s, \tag{2.2}$$

$$y_1 = y_0 + h \sum_{j=1}^{s} b_j f(x_0 + c_j h, Y_j), \tag{2.3}$$

est appelée méthode de Runge-Kutta à s étapes. Les Y_i sont dits approximations internes.

Remarques :

- Si A est triangulaire supérieure ou inférieure stricte, la méthode est dite explicite. Les approximations internes peuvent être calculées l'une après l'autre de manière explicite.
- Dans le cas contraire, la méthode est dite implicite. La détermination des approximations internes nécessite alors la résolution d'un système généralement non-linéaire à $s \times m$ inconnues, si m est la dimension du système différentiel (1.1).
- On représente une méthode de Runge-Kutta par son *tableau de Butcher* (J.C. Butcher, 1964),

$$\begin{array}{c|ccc} c_1 & a_{11} & \dots & a_{1s} \\ \vdots & \vdots & & \vdots \\ c_s & a_{s1} & \dots & a_{ss} \\ \hline & b_1 & \dots & b_s \end{array} . \tag{2.4}$$

  On a par exemple les tableaux suivants respectivement pour les méthodes d'Euler explicite, Euler implicite et de Runge :

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}, \quad \begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}, \quad \begin{array}{c|cc} 0 & 0 & 0 \\ 1/2 & 1/2 & 0 \\ \hline & 0 & 1 \end{array} .$$

2.3 Étude de la convergence

L'erreur globale désigne l'erreur de la solution approchée après un «certain nombre» de pas. On l'estime ici dans le cadre plus général des méthodes à un pas, c'est-à-dire de la forme

$$y_1 = y_0 + h\Phi(x_0, y_0, h), \tag{2.5}$$

où Φ est une fonction continue de $[x_0, X] \times \mathbb{R}^m \times [0, h^*]$, $h^* > 0$, à valeurs dans $\mathbb{R}^m$, qui ne dépend que de la fonction f du problème (1.1). Par exemple, la méthode d'Euler explicite est la méthode à un pas qui correspond à la fonction

$$\Phi(x, y, h) = f(x, y).$$

Étant donnée une subdivision x_0, x_1, ..., $x_N = X$ de l'intervalle $[x_0, X]$, la solution numérique au point X est obtenue par applications successives de l'équation

$$y_{i+1} = y_i + h_i\Phi(x_i, y_i, h_i), \quad h_i = x_{i+1} - x_i. \tag{2.6}$$

L'objectif de ce paragraphe est donc de borner l'erreur après N pas

$$E = y(X) - y_N. \tag{2.7}$$

Cette borne est obtenue en remarquant que E résulte du transport et de l'accumulation des erreurs locales. En effet, si pour chaque i on note $y^{[i]}(x)$ la solution exacte de $y'(x) = f(x, y(x))$ sur $[x_i, X]$ telle que $y(x_i) = y_i$, il vient

$$\begin{aligned} E &= y(X) - y_N \\ &= \Big(y(X) - y^{[1]}(X)\Big) + \sum_{k=1}^{N-2} \Big(y^{[k]}(X) - y^{[k+1]}(X)\Big) + \Big(y^{[N-1]}(X) - y_N\Big) \\ &= E_1 + \sum_{k=1}^{N-2} E_{k+1} + E_N, \end{aligned} \tag{2.8}$$

et chacun des termes E_i de cette somme correspond à la contribution de l'erreur locale $e_i = y^{[i-1]}(x_{i-1}+h_{i-1})-y_i$ à l'erreur globale. Le processus est représenté sur la figure 2.2. On supposera donc, qu' à l'instar de la majoration (2.1), on a :

$$\|e_i\| \leq Ch_{i-1}^{p+1}. \tag{2.9}$$

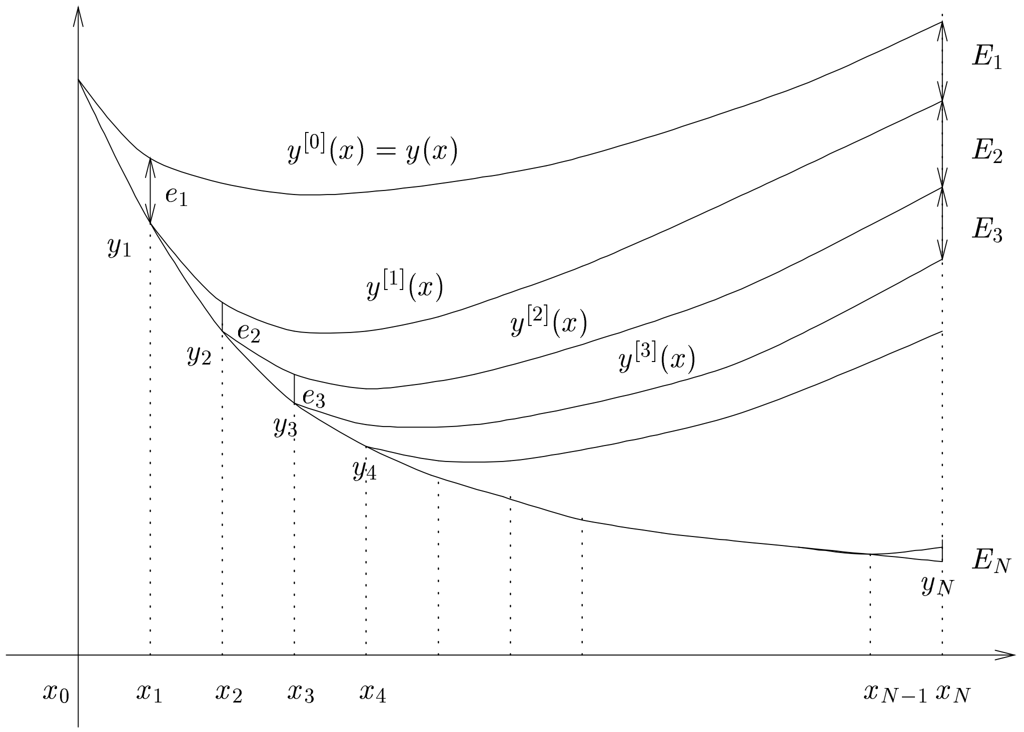


FIG. 2.2: *Transport des erreurs*

Lemme 1 (dit lemme fondamental) *Soit $y(x)$ la solution exacte du système différentiel (1.1), et soit $v(x)$ une solution approchée, supposée continuement différentiable sur $[x_0, X]$, telle que*

$$\|v'(x) - f(x, v(x))\| \leq \epsilon, \tag{2.10}$$
$$\|y(x_0) - v(x_0)\| \leq \rho, \tag{2.11}$$
$$\|f(x, v) - f(x, y)\| \leq L\|v - y\|. \tag{2.12}$$

Alors, pour tout x de $[x_0, X]$, on a

$$\|y(x) - v(x)\| \leq \rho e^{L(x-x_0)} + \frac{\epsilon}{L}\left(e^{L(x-x_0)} - 1\right). \tag{2.13}$$

démonstration : la démonstration complète est donnée dans [13], pp. 58. Pour $\epsilon = 0$, et $\|.\| = \|.\|_2$, il vient

$$\begin{aligned}
\frac{d\|y(x) - v(x)\|_2^2}{dx} &= 2(y(x) - v(x))^T \left(f(x, y(x)) - f(x, v(x))\right), \\
&\leq 2L\|y(x) - v(x)\|_2^2.
\end{aligned}$$

En posant $\mu(x) = \|y(x) - v(x)\|_2^2$, on a donc :

$$\frac{d\left(e^{-2L(x-x_0)}\mu(x)\right)}{dx} \leq 0.$$

Il résulte que

$$\mu(x) \leq \rho^2 e^{2L(x-x_0)}.$$

Quelque soit la norme, on a maintenant

$$\|y(x) - v(x)\| \leq C\rho e^{L(x-x_0)}.$$

□

Théorème 1 *Soit U un voisinage de $\{(x, y(x)), x_0 \leq x \leq X\}$ où $y(x)$ désigne la solution exacte de (1.1). Supposons que f soit Lipschitzienne dans U de constante de Lipschitz L et que la majoration (2.9) soit correcte dans U. Alors, on a*

$$\|E\| \leq h^p \frac{C}{L}\left(e^{L(x_N - x_0)} - 1\right), \tag{2.14}$$

avec $h = \max_i h_i$.

démonstration : en prenant $\epsilon = 0$ et $\rho = \|e_i\|$ dans le lemme fondamental, (2.8) devient

$$\begin{aligned}
\|E\| &\leq Ch^p \sum_{i=1}^{N} h_{i-1} e^{L(X-x_i)} \\
&\leq Ch^p \int_{x_0}^{X} e^{L(X-x)} dx \\
&\leq h^p \frac{C}{L}\left(e^{L(X-x_0)} - 1\right).
\end{aligned}$$

□

2.4 Étude de l'erreur locale

Le paragraphe précédent a permis de quantifier la propagation des erreurs locales. La construction effective de schémas nécessite désormais que l'on s'intéresse plus précisément aux termes de la forme

$$e = y(x_0 + h) - y_1, \tag{2.15}$$

où y_1 désigne comme jusqu'alors la solution numérique après un pas. L'objectif est ici de majorer $\|e\|$ par une quantité de la forme Ch^{p+1}, conformément à l'hypothèse (2.9). p est alors, par définition, l'ordre de la méthode. Il détermine la précision asymptotique de la méthode. La technique mise en oeuvre ci-après fait appel à une représentation en arbres des développements en série de Taylor de $y(x_0 + h)$ et y_1. Leur comparaison terme à terme permet alors la détermination de p en fonction des coefficients de la méthode A, b, et c. Pour des raisons de simplicité, on suppose que f ne dépend ici que de y. Il est aisé de voir, en introduisant une nouvelle variable $y_{m+1} = x$ dans (1.1), que cette hypothèse ne nuit pas à la généralité.

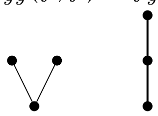
2.4.1 Dérivées de la solution exacte

La solution numérique ne pouvant être développée qu'en fonction des dérivées de f, il convient d'exprimer $y'(x)$, $y''(x)$, ..., suivant ces mêmes expressions. Les premiers termes du développement à l'ordre 4 de $y(x_0 + h)$ s'écrivent par exemple (en omettant les arguments x_0 et $y(x_0)$) :

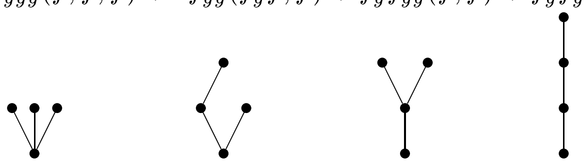
$$y' = f(y)$$

$$y'' = \frac{\partial f}{\partial y} f = f_y f$$

$$y^{(3)} = f_{yy}(f,f) + f_y f_y f$$



$$\begin{aligned} y^{(4)} &= f_{yyy}(f,f,f) + f_{yy}(f_y f, f) + f_{yy}(f, f_y f) + f_{yy}(f_y f, f) + f_y f_{yy}(f,f) + f_y f_y f_y f \\ &= f_{yyy}(f,f,f) + 3 f_{yy}(f_y f, f) + f_y f_{yy}(f,f) + f_y f_y f_y f \end{aligned}$$



En dessous de chaque élément différentiel f, $f_y f$, ..., a été associée la représentation graphique d'un arbre, dont l'interprétation peut être décrite en 3 points :

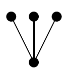
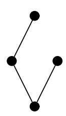
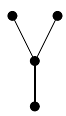
- la racine de l'arbre, comme tous les noeuds, correspond à l'élément différentiel obtenu en dérivant f un certain nombre de fois (par rapport à y);
- si un noeud possède n branches, alors f est dérivée n fois. C'est une fonction multilinéaire à n arguments;
- le $i^{\text{ème}}$ argument de cette fonction est l'élément différentiel correspondant à la $i^{\text{ème}}$ branche issue de ce noeud.

L'équivalence entre arbres et éléments différentiels peut maintenant être décrite formellement :

Définition 2 *L'ensemble T des arbres est défini récursivement par :*

1. $\emptyset \in T$.
2. $\tau \in T$.
3. $t = [t_1, \dots, t_n] \in T$, *ssi* $t_1, \dots, t_n \in T$.

La notation $[t_1, \dots, t_n]$ désigne l'arbre obtenu en connectant les n branches $t_1, \dots, t_n$ à une nouvelle racine commune. L'ordre des branches n'importe pas, pas plus que l'ordre des arguments de $f_{yy}(f_y f, f)$, ou $f_{yyy}(f, f_y, f)$. Par exemple, on notera

$[\tau, \tau, \tau] =$  , $[[\tau], \tau] =$  , $[[\tau, \tau]] =$  , $[[[\tau]]] =$  .

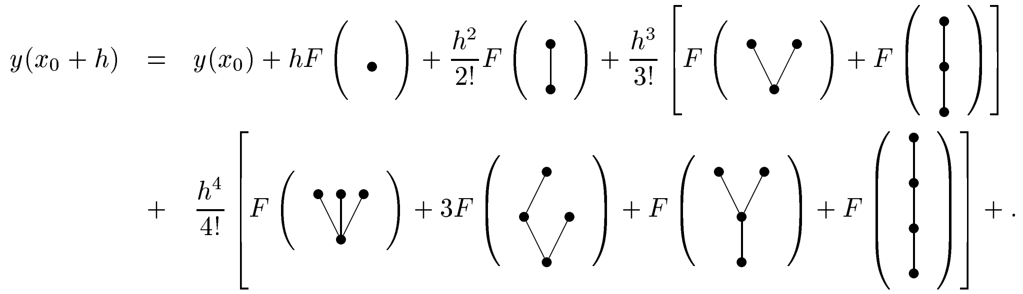
Définition 3 *A tout arbre t de T, on associe l'élément différentiel $F(t)$ défini récursivement par :*

1. $F(\emptyset) = y$,
2. $F(\tau) = f$,
3. $F([t_1, \dots, t_n]) = \frac{\partial^n f}{\partial y^n}\,(F(t_1), \dots, F(t_n))$.

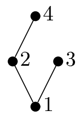
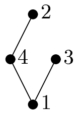
La solution exacte de (1.1) au point $x_0 + h$ peut alors s'écrire sous la forme

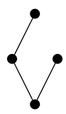
$$\begin{aligned} y(x_0+h) &= y(x_0) + h\,(f)_{y=y_0} + \frac{h^2}{2!}\,(f_y f)_{y=y_0} + \frac{h^3}{3!}\,(f_{yy}(f,f) + f_y f_y f)_{y=y_0} \\ &+ \frac{h^4}{4!}\,(f_{yyy}(f,f,f) + 3f_{yy}(f_y f, f) + f_y f_{yy}(f,f) + f_y f_y f_y f)_{y=y_0} + \dots, \end{aligned}$$

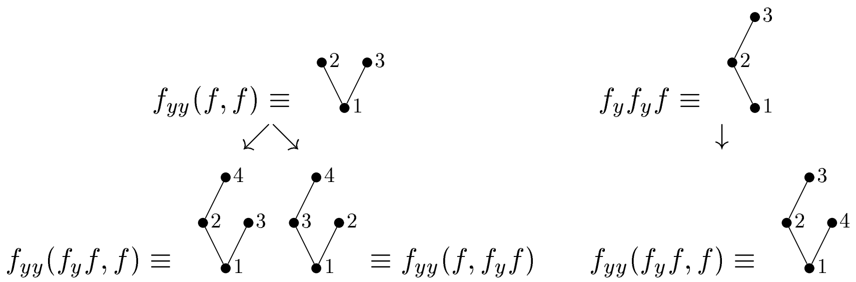
ou de façon équivalente

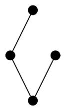


(2.16)

Si l'ordre $\rho(t)$ d'un arbre t est défini comme son nombre de noeuds, on peut vérifier que les arbres d'ordre q apparaissent tous dans $y^{(q)}(x_0+h)$. Il reste finalement à déterminer le nombre de fois $\alpha(t)$ que chaque arbre t apparaît. Le comptage s'effectue par récurrence en introduisant un processus d'*indiçage* des arbres (voir [6], pp. 92 ou [13], pp.147). L'indiçage est dit monotone lorsque les indices croissent le long des branches, en partant de la racine. Ainsi,  est monotone alors que  ne l'est pas. L'ordre des indices correspond de façon naturelle à l'ordre de différentiation.

Par exemple, l'élément différentiel associé à  est obtenu en différentiant les arguments de l'élément différentiel $f_{yy}(f,f)$ (ce qui peut être fait de deux façons) ou en différentiant le premier terme de l'élément différentiel $f_y f_y f$:



On a donc $\alpha($  $)=3$.

Théorème 2 *Soit $t \in T$. Il peut être représenté sous la forme*

$$t = [\underbrace{t_1, \dots, t_1}_{m_1}, \underbrace{t_2, \dots, t_2}_{m_2}, \dots, \underbrace{t_n, \dots, t_n}_{m_n}], \tag{2.17}$$

où tous les t_i sont supposés distincts. On a alors

$$\begin{aligned} &\alpha(\emptyset) = 1, \\ &\alpha(\tau) = 1, \\ &\alpha(t) = (\rho(t) - 1)! \prod_{i=1}^{n} \frac{1}{m_i!} \left(\frac{\alpha(t_i)}{\rho(t_i)!} \right)^{m_i}. \end{aligned}$$

démonstration : le résultat est évident pour $t = \emptyset$ et pour $t = \tau$. Supposons qu'il soit vrai pour tous le arbres d'ordres inférieurs à $q-1$ et considérons une arbre t de la forme (2.17) d'ordre q. On dispose de q indices dont le plus petit est nécessairement affecté à la racine. Supposons choisis $m_i\rho(t_i)$ indices parmi les $q-1$ indices restants. On peut alors indicer les m_i branches t_i de t de

$$\frac{\alpha(t_i)^{m_i}}{m_i!} \binom{\rho(t_i)}{m_i\rho(t_i)} \binom{\rho(t_i)}{(m_i-1)\rho(t_i)} \binom{\rho(t_i)}{(m_i-2)\rho(t_i)} \cdots \binom{\rho(t_i)}{\rho(t_i)}$$

manières différentes. C'est-à-dire plus simplement,

$$\frac{(m_i\rho(t_i))!}{m_i!} \left(\frac{\alpha(t_i)}{\rho(t_i)!} \right)^{m_i}.$$

Or il y a

$$\binom{m_1\rho(t_1)}{q-1} \binom{m_2\rho(t_2)}{q-1-m_1\rho(t_1)} \binom{m_3\rho(t_3)}{q-1-m_1\rho(t_1)-m_2\rho(t_2)} \cdots \binom{m_n\rho(t_n)}{m_n\rho(t_n)}$$

manières différentes de partitionner $q-1$ indices en n parties à $m_1\rho(t_1)$, $m_2\rho(t_2)$, ..., $m_n\rho(t_n)$ indices. Ou plus simplement là encore,

$$\frac{(q-1)!}{\prod_i^n (m_i\rho(t_i))!}.$$

Le nombre $\alpha(t)$ cherché est donc

$$\frac{(q-1)!}{\prod_i^n (m_i\rho(t_i))!} \prod_{i=1}^{n} \frac{(m_i\rho(t_i))!}{m_i!} \left(\frac{\alpha(t_i)}{\rho(t_i)!} \right)^{m_i} = (\rho(t)-1)! \prod_{i=1}^{n} \frac{1}{m_i!} \left(\frac{\alpha(t_i)}{\rho(t_i)!} \right)^{m_i}.$$

□.

A titre d'exemples, $\alpha([[\tau],\tau]) = 3$, $\alpha([[\tau],\tau,\tau]) = 6$, ou $\alpha([[[\tau]],\tau]) = 4$. L'ensemble des résultats de ce paragraphe donne l'expression du développement recherché :

Théorème 3 *La solution exacte de (1.1) s'écrit formellement*

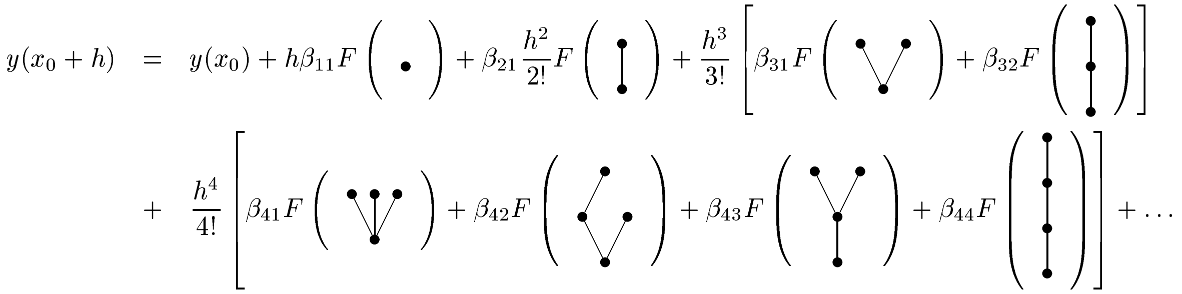
$$y(x_0 + h) = \sum_{i=0}^{\infty} \frac{h^i}{i!} \sum_{t \in T,\ \rho(t)=i} \alpha(t) F(t).$$

2.4.2 Dérivées de la solution numérique

La solution approchée y_1 au point $x_0 + h$ fournie par une méthode de Runge-Kutta s'écrit elle aussi sous la forme

$$\begin{aligned} y_1 &= y(x_0) + h\left(\beta_{11} f\right)_{y=y_0} + \frac{h^2}{2!}\left(\beta_{21} f_y f\right)_{y=y_0} + \frac{h^3}{3!}\left(\beta_{31} f_{yy}(f,f) + \beta_{32} f_y f_y f\right)_{y=y_0} \\ &+ \frac{h^4}{4!}\left(\beta_{41} f_{yyy}(f,f,f) + \beta_{42} f_{yy}(f_y f, f) + \beta_{43} f_y f_{yy}(f,f) + \beta_{44} f_y f_y f_y f\right)_{y=y_0} + \dots, \end{aligned}$$

ou de façon équivalente



Les coefficients β dépendent de la méthode considérée et peuvent être évalués directement à partir des arbres auxquels ils sont attachés. On a besoin pour cela de deux fonctions Φ et γ, définies sur T.

Définition 4 *On appelle fonction de poids la fonction définie sur T récursivement par :*

1. $\Phi(\emptyset) = A^{-1}e,\ e = (1, \dots, 1)^T \in \mathbb{R}^s$,
2. $\Phi(\tau) = e$,
3. $\Phi([t_1, \dots, t_n]) = \prod_{i=1}^{n} A\Phi(t_i)$,

où A est la matrice de coefficients de la méthode considérée.

Φ est une fonction à valeur dans $\mathbb{R}^s$. Ici, le produit de deux vecteurs de $\mathbb{R}^s$ est donc à entendre composante par composante.

Définition 5 *On appelle fonction densité la fonction définie sur T récursivement par :*

1. $\gamma(\emptyset) = 1$,
2. $\gamma(\tau) = 1$,
3. $\gamma([t_1, \dots, t_n]) = \rho(t) \prod_{i=1}^{n} \gamma(t_i)$.

Exemples :

1. $t = [\tau, \tau, \tau] : \Phi(t) = (Ae).(Ae).(Ae)$ et $\gamma(t) = 4 \times 1 \times 1 \times 1$.
2. $t = [[\tau, \tau], \tau] : \Phi(t) = (Ae).\left(A[(Ae).(Ae)]\right)$ et $\gamma(t) = 5 \times [1 \times 3 \times (1 \times 1)] = 15$.

On peut alors montrer (voir [13] ou [6]) le théorème suivant :

Théorème 4 *Soit $\mathcal{R}$ une méthode de Runge-Kutta à s étapes définie par les vecteurs b et c et la matrice A. La solution approchée y_1 obtenue par $\mathcal{R}$ après un pas s'écrit formellement*

$$y_1 = \sum_{i=0}^{\infty} \frac{h^i}{i!} \sum_{t \in T,\ \rho(t)=i} \alpha(t) \left(\gamma(t) b^T \Phi(t)\right) F(t) \tag{2.18}$$

2.4.3 Conditions d'ordre

En comparant les théorèmes 3 et 4 on obtient finalement :

Théorème 5 *Une méthode de Runge-Kutta est d'ordre local $p+1$ (c'est-à-dire d'ordre global p) si et seulement si*

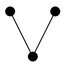
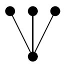
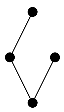
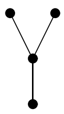
$$\forall t \in T \,/\, \rho(t) \leq p,\ b^T\Phi(t) = \frac{1}{\gamma(t)}. \tag{2.19}$$

2.4.4 Exemples de construction

A titre d'illustration, on cherche ici à construire une méthode d'ordre 4. Soit donc $\mathcal{R} = (A, b, c)$, une méthode à s étapes. On rappelle que

$$A \in \mathbb{R}^{s\times s},\ b \in \mathbb{R}^s,\ c \in \mathbb{R}^s.$$

Les conditions d'ordre sont obtenues en construisant les arbres d'ordre inférieur ou égal à 4 :

Ordre	Graphe	$\gamma(t)$	$\Phi(t)$	Condition d'ordre
1	$t =$	$\gamma(t) = 1$	$\Phi(t) = e$	$b^T e = 1$
2	$t =$ 	$\gamma(t) = 2$	$\Phi(t) = Ae$	$b^T Ae = 1/2$
3	$t =$ 	$\gamma(t) = 3$	$\Phi(t) = (Ae).(Ae)$	$b^T(Ae)^2 = 1/3$
3	$t =$ 	$\gamma(t) = 6$	$\Phi(t) = A^2e$	$b^T A^2 e = 1/6$
4	$t =$ 	$\gamma(t) = 4$	$\Phi(t) = (Ae)^3$	$b^T(Ae)^3 = 1/4$
4	$t =$ 	$\gamma(t) = 8$	$\Phi(t) = (Ae).(A^2e)$	$b^T[(Ae).(A^2e)] = 1/8$
4	$t =$ 	$\gamma(t) = 12$	$\Phi(t) = A(Ae)^2$	$b^T A(Ae)^2 = 1/12$
4	$t =$ 	$\gamma(t) = 24$	$\Phi(t) = A^3e$	$b^T A^3 e = 1/24$

Il est clair que $s = 1$ ne permet pas de satisfaire toutes les conditions d'ordre. Pour $s = 2$, et en supposant que $Ae = c$, il est aisé de vérifier que $c_1 = c_2$ ne convient pas. On suppose donc que c_1 et c_2 sont distincts de sorte que e, c forment une base de $\mathbb{R}^2$. Il existe donc δ_1 et δ_2, tels que

$$c^2 = \delta_1 e + \delta_2 c.$$

Or, $b^T[e, c, c^2, c^3] = (1, 1/2, 1/3, 1/4)$, donc

$$\begin{cases} \delta_1 + \frac{1}{2}\delta_2 = \frac{1}{3} \\ \delta_1\delta_2 + \frac{1}{2}(\delta_1 + \delta_2^2) = \frac{1}{4} \end{cases},$$

et on obtient $\delta_1 = -1/6$ et $\delta_2 = 1$. c_1 et c_2 sont donc les racines de $x^2 - x + 1/6 = 0$. On prend par exemple $c_1 = \frac{1}{2} - \frac{\sqrt{3}}{6}$ et $c_2 = \frac{1}{2} + \frac{\sqrt{3}}{6}$. Le système $b^T[e, c] = (1, 1/2)$ donne alors $b_1 = b_2 = 1/2$. De même, il existe μ_1 et μ_2, tels que

$$Ac = \mu_1 e + \mu_2 c.$$

Or, $b^T[Ac, A^2c] = (1/6, 1/24)$, donc

$$\begin{cases} \mu_1 + \frac{1}{2}\mu_2 = \frac{1}{6} \\ \mu_1\mu_2 + \frac{1}{2}(\mu_1 + \mu_2^2) = \frac{1}{24} \end{cases},$$

et on obtient $\mu_1 = -1/12$ et $\mu_2 = 1/2$. Ce qui signifie en particulier que

$$Ac = \frac{1}{2}c^2.$$

Réciproquement, si $Ac = \frac{1}{2}c^2$, alors on peut vérifier que

$$\begin{aligned} b^T Ac &= \frac{1}{2}b^T c^2 = \frac{1}{6}, \\ b^T Ac^2 &= b^T A(c - \frac{1}{2}e) = b^T Ac - \frac{1}{6}b^T Ae = \frac{1}{12}, \\ b^T A^2 c &= \frac{1}{2}b^T Ac^2 = \frac{1}{24}, \\ b^T(c.Ac) &= \frac{1}{2}b^T(c.c^2) = \frac{1}{2}b^T c^3 = \frac{1}{6}. \end{aligned}$$

Il n'existe donc qu'une seule méthode d'ordre 4 avec $s = 2$. C'est la méthode de *Gauss* :

$$\begin{array}{c|cc} \frac{1}{2} - \frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4} - \frac{\sqrt{3}}{6} \\ \frac{1}{2} + \frac{\sqrt{3}}{6} & \frac{1}{4} + \frac{\sqrt{3}}{6} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

Cherchons désormais à construire une méthode explicite d'ordre 3. Il est aisé de constater que $s = 2$ est insuffisant. On prend donc $s = 3$. En supposant comme précédemment que $Ae = c$, A et c sont de la forme

$$A = \begin{bmatrix} 0 & 0 & 0 \\ c_2 & 0 & 0 \\ a_{31} & c_3 - a_{31} & 0 \end{bmatrix} \text{ et } c = \begin{bmatrix} 0 \\ c_2 \\ c_3 \end{bmatrix}.$$

Des conditions $b^T e = 1$, $b^T Ae = 1/2$ et $b^T(Ae)^2 = 1/3$ on tire aisément b_1, b_2 et b_3. a_{31} est alors obtenu à partir de l'équation restante $b^T A^2 e = 1/6$. On obtient alors la méthode suivante

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ c_2 & c_2 & 0 & 0 \\ c_3 & \frac{c_3(3c_2^2 - 3c_2 + c_3)}{c_2(3c_2 - 2)} & \frac{c_3(c_2 - c_3)}{c_2(3c_2 - 2)} & 0 \\ \hline & \frac{1}{6}\frac{-3c_2 + 6c_2c_3 + 2 - 3c_3}{c_2c_3} & \frac{-1}{6}\frac{3c_3 - 2)}{c_2(c_2 - c_3)} & \frac{1}{6}\frac{3c_2 - 2)}{c_2(c_2 - c_3)} \end{array}$$

On peut par exemple éliminer les paramètres libres restants en imposant les deux conditions d'ordre 4, $b^T(Ae)^3 = 1/4$ et $b^T A(Ae)^2 = 1/12$, ce qui donne finalement la méthode

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1 & -1 & 2 & 0 \\ \hline & 1/6 & 2/3 & 1/6 \end{array}.$$

2.5 Stabilité des méthodes de Runge-Kutta

Le coût calcul d'une méthode de Runge-Kutta est essentiellement déterminé par la valeur du paramètre s, nombre d'étapes internes, et par son caractère explicite ou implicite. Pour une méthode explicite, chaque pas d'intégration nécessite $(s-1)$ évaluations du second membre f du système à résoudre (1.1). Pour une méthode implicite, chaque pas exige la résolution d'un système généralement non-linéaire à $s \times m$ inconnues, si m est la dimension du système (1.1). Une rapide évaluation du nombre d'opérations nécessaires dans chacun des deux cas fait apparaître un déséquilibre important. A-priori, il semble donc superfétatoire de chercher à paralléliser des méthodes aussi lourdement handicapées que les méthodes de Runge-Kutta implicites. L'exemple suivant,

$$\begin{cases} y'(x) = \lambda y(x) \\ y(0) = y_0 \end{cases}, \tag{2.20}$$

dû à G. Dahlquist, montre pourtant l'avantage qu'il peut y avoir à recourir à une méthode implicite. Il est clair que $y(x) = e^{\lambda x} y_0$ est la solution exacte de (2.20), quelque soit la valeur de λ. Pour λ à partie réelle négative, la solution exacte est donc asymptotiquement nulle. Par ailleurs, la solution numérique obtenue par une méthode de Runge-Kutta pour un pas h est donnée par l'équation de récurrence

$$y_{n+1} = R(\lambda h) y_n,$$

où $R(z) = 1 + zb^T(I - zA)^{-1}e$.

Définition 6 *La fonction $R(z)$ est appelée fonction de stabilité de la méthode. Elle peut être interprétée comme la solution numérique après un pas pour le problème test*

$$y' = \lambda y, \; y_0 = 1, \; z = \lambda h.$$

L'ensemble

$$S = \{z \in \mathbb{C}; |R(z)| \leq 1\}$$

est appelé domaine de stabilité de la méthode.

Lorsque la méthode est explicite, $R(z)$ est un polynôme de degré inférieur ou égal à s. En effet, A est alors nilpotente d'indice de nilpotence inférieur ou égal à s, de sorte que

$$\begin{aligned} R(z) &= 1 + zb^T(I + zA + z^2A^2 + \cdots + z^{s-1}A^{s-1})e, \\ &= 1 + (b^Te)z + (b^TAe)z^2 + \cdots + (b^TA^{s-1}e)z^s. \end{aligned}$$

En conséquence, le domaine de stabilité S est borné. En fait, S est essentiellement caractérisé par le théorème suivant

Théorème 6 *Soit $\mathcal{R} = (A, b, c)$ une méthode de Runge-Kutta d'ordre p. Alors,*

$$R(z) = 1 + z + \frac{1}{2}z^2 + \cdots + \frac{1}{p!}z^p + \mathcal{O}(z^{p+1}).$$

démonstration : la solution exacte du problème test étant e^z, on doit nécessairement avoir

$$R(z) - e^z = \mathcal{O}(h^{p+1}) = \mathcal{O}(z^{p+1}).$$

$\square$.

Pour les valeurs de $z = \lambda h$ en dehors S, y_n tend vers l'infini lorsque n tend vers l'infini. Pour espérer approcher le comportement de la solution exacte, il est donc nécessaire d'utiliser un pas h tel que $\lambda h \in S$. C'est une restriction sévère pour les grandes valeurs de λ. Cette restriction peut être relaxée voir totalement levée pour les méthodes implicites, pour lesquelles $R(z)$ devient une fraction rationnelle en z de la forme $\frac{P(z)}{Q(z)}$, $d^oP \leq s$, $d^oQ \leq s$. A titre d'exemple, les méthodes d'Euler explicite et implicite ont respectivement $R(z) = 1 + z$ et $R(z) = \frac{1}{1-z}$ comme fonctions de stabilité. Il est donc aisé de voir que S est le disque de centre -1 et de rayon 1 pour Euler explicite alors qu'il s'étend à tout le demi-plan à gauche de l'axe des imaginaires pour Euler implicite. Ainsi, si l'on prend $\lambda = -20$ dans (2.20), la méthode d'Euler explicite ne donne des résultats acceptables que pour $h < 0.1$. Dans ce type de situations, les méthodes implicites peuvent donc s'avérer plus performantes. Lorsque le domaine de stabilité S d'une méthode contient le demi-plan complexe gauche, -comme c'est le cas de la méthode d'Euler implicite-, le comportement de la solution numérique pour le problème (2.20) coïncide alors avec celui de la solution exacte.

Définition 7 *Une méthode dont le domaine de stabilité S contient $\mathbb{C}^- = \{z; Re(z) < 0\}$ est dite A-stable.*

Il est clair qu'une méthode explicite ne peut être A-stable. En fait, -et en raison du principe du

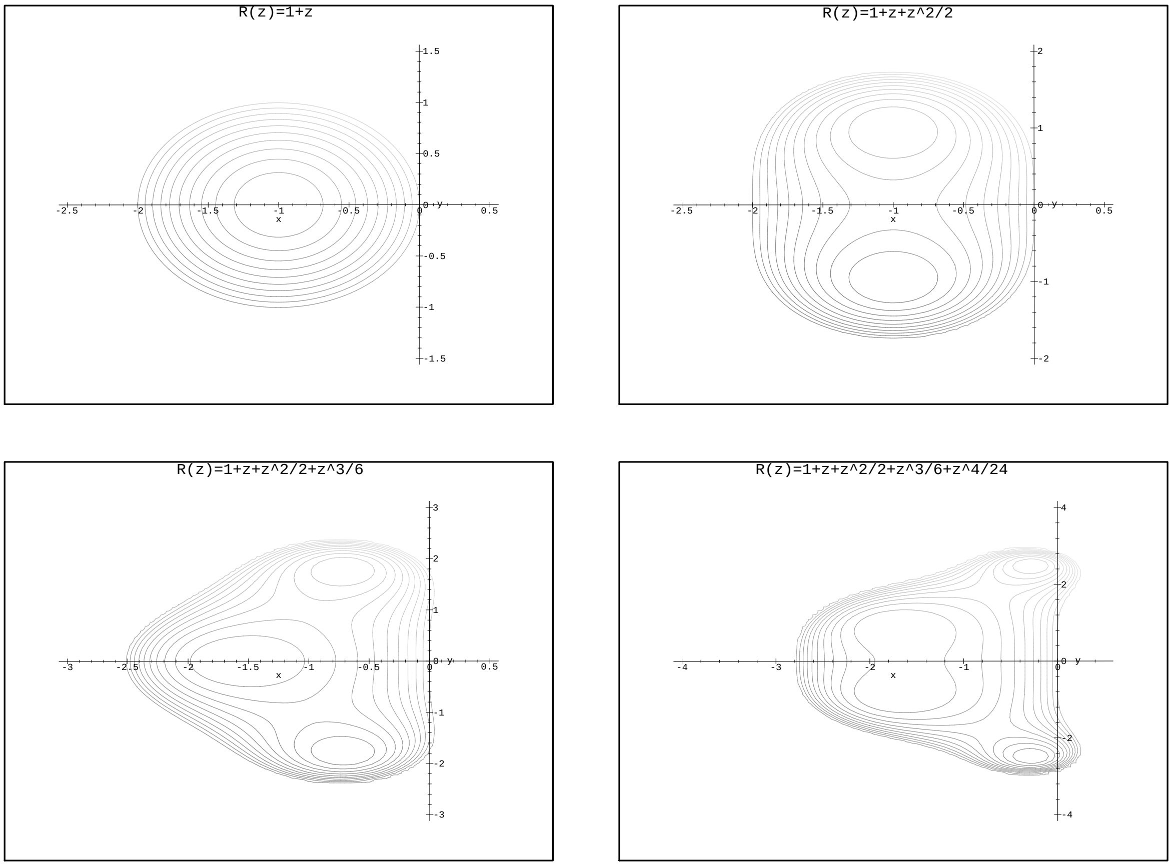


FIG. 2.3: *Domaines de stabilité*

maximum-, une méthode est A-stable si et seulement si $|R(iy)| \leq 1$ pour tout $y \in \mathbb{R}$ et $R(z)$ est analytique pour $Re(z) \leq 0$. L'étude de la A-stabilité des approximations rationnelles de $\exp(z)$ a conduit au développement de la théorie dite des étoiles, fondée en 1978 par G. Wanner, E. Hairer et S. Nørsett (voir [14] ou [15]). Son objet est non pas l'étude directe du domaine de stabilité mais de l'ensemble $\{z \in \mathbb{C};\, \frac{|R(z)|}{|\exp(z)|} > 1\}$, dit «étoile de R» (voir figure 2.4).

Exemples : les méthodes d'Euler implicite et de Gauss d'ordre 4 sont A-stables. En effet, on a

$$R_E = \frac{1}{1-z},$$
$$R_G = \frac{1+z/2+z^2/12}{1-z/2+z^2/12}.$$

Il est immédiat que $|R_E(z)| < 1$ pour $Rez < 0$. Quant à R_G, il vient

$$\forall y \in \mathbb{R},\ |1 - iy/2 + (iy)^2/12|^2 - |1 + iy/2 + (iy)^2/12|^2 = 0 \geq 0.$$

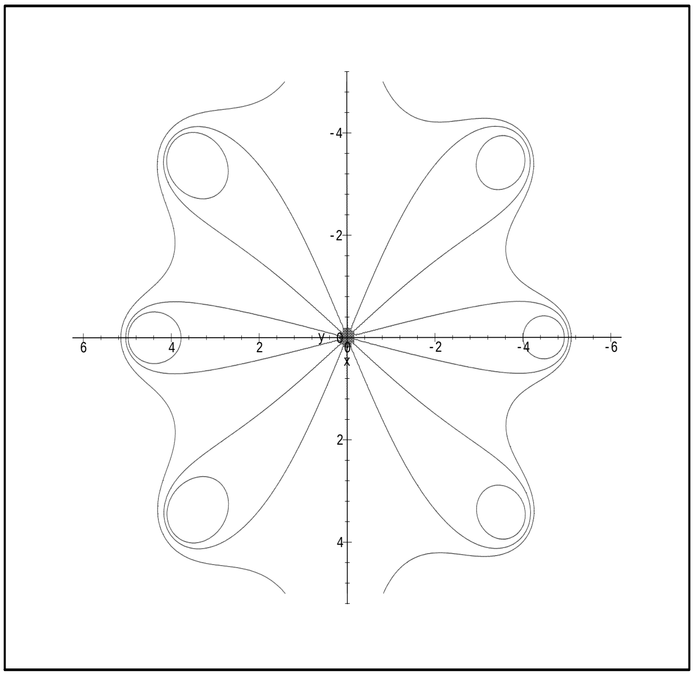


FIG. 2.4: *Étoile de* $R(z) = \frac{1+z/2+z^2/10+z^3/120}{1-z/2+z^2/10-z^3/120}$ *(méthode de Gauss d'ordre 6).*

Chapitre 3

Implémentation parallèle des méthodes de Runge-Kutta

3.1 Parallélisation des méthodes explicites

Hormis donc la parallélisation à travers les équations, dont les limites apparaissent pour les systèmes de petite taille, il est possible d'exhiber un certain degré de parallélisme à travers la méthode. Il s'agit ici d'évaluer le second membre f en différents points de manière concurrente. Ainsi, par exemple, la méthode d'ordre 2 suivante

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \hline & 1/6 & 2/3 & 1/6 \end{array},$$

est telle qu'il est possible de calculer $f(Y_2)$ et $f(Y_3)$ au même instant. D'une façon générale, si la matrice A peut être partitionnée en

$$A = \begin{pmatrix} 0 & & & & \\ A_{21} & 0 & & & \\ A_{31} & A_{32} & \ddots & & \\ \vdots & \vdots & & \ddots & \\ A_{\sigma 1} & A_{\sigma 2} & \dots & A_{\sigma,\sigma-1} & 0 \end{pmatrix},$$

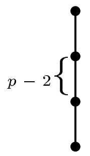
où $A_{i,j}$ est de taille $\mu_i \times \mu_j$, alors $f(Y_1)$, ..., $f(Y_{\mu_1})$ peuvent être calculés simultanément, puis $f(Y_{\mu_1+1})$, ..., $f(Y_{\mu_1+\mu_2})$, etc. Ainsi, si l'on dispose de $\mu = \max \mu_i$ processeurs, la méthode nécessite σ évaluations séquentielles de f. Malheureusement, le théorème suivant, dû à K. Jackson et S. Nørsett, donne une majoration sévère de l'ordre d'une méthode parallèle.

Théorème 7 *Une méthode de Runge-Kutta explicite à σ étapes séquentielles est d'ordre inférieur ou égal à σ, quel que soit le nombre de processeurs disponibles.*

démonstration : soit p l'ordre de la méthode. D'après le théorème 5 on a alors

$$\forall\, t \in T\,/\rho(t) \leq p,\ b^T\Phi(t) = \frac{1}{\gamma(t)}.$$

Il résulte que $\Phi(t) \neq 0$ pour $t = \underbrace{[\dots[}_{p-1}\tau\underbrace{]\dots]}_{p-1}$, c'est-à-dire pour l'arbre d'ordre p de plus grande «taille»



Or $\Phi(t) = A^{p-1}e$, donc il existe au moins un indice i_1, $1 \leq i_1 \leq s$ tel que

$$\sum_{j_2,\dots,j_p=0}^{s} a_{i_1j_2}a_{j_2j_3}\dots a_{j_{p-1}j_p} \neq 0.$$

En particularisant davantage encore, on peut affirmer qu'il existe p indices $i_1,\dots,i_p$ tels que

$$a_{i_1i_2}a_{i_2i_3}\dots a_{i_{p-1}i_p} \neq 0. \tag{3.1}$$

La méthode étant explicite, A est triangulaire stricte, -par exemple triangulaire inférieure stricte-, c'est-à-dire telle que $a_{ij} = 0$ pour $j \geq i$. Pour que $a_{i_1i_2}a_{i_2i_3}\dots a_{i_{p-1}i_p}$ soit non nul, il faut donc que $i_1 > i_2 > i_3 \dots > i_{p-1} > i_p$, ce qui implique que $f(Y_{i_1})$ soit calculé après $f(Y_{i_2})$, $f(Y_{i_2})$ après $f(Y_{i_3})$, ..., $f(Y_{i_{p-1}})$ après $f(Y_{i_p})$. Donc $\sigma \geq p$. □

Les méthodes telles que $\sigma = p$ sont optimales dans un environnement parallèle. Pour $p \leq 4$, la construction de méthodes d'ordre p est possible avec $\sigma = s = p$. Ainsi, les méthodes dites «classique» et «des 3/8»

$$\begin{array}{c|cccc} 0 & & & & \\ 1/2 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ 1 & 0 & 0 & 1 & \\ \hline & 1/6 & 2/6 & 2/6 & 1/6 \end{array} \qquad \begin{array}{c|cccc} 0 & & & & \\ 1/3 & 1/3 & & & \\ 2/3 & -1/3 & 1 & & \\ 1 & 1 & -1 & 1 & \\ \hline & 1/8 & 3/8 & 3/8 & 1/8 \end{array},$$

sont optimales. Le recours au parallélisme ne se justifie donc que pour les ordres supérieurs à 4.

3.1.1 Exemple de construction d'une méthode d'ordre 5 avec $\sigma = 5$ et $s = 6$

En plus des conditions d'ordre énumérées au paragraphe 2.4.4 pour l'ordre 4, 9 nouvelles conditions doivent être satisfaites à l'ordre 5. En supposant là encore que $Ae = c$, ces conditions s'écrivent :

$$\begin{aligned} &b^Tc^4 = \frac{1}{5}, b^Tc^2.Ac = \frac{1}{10}, b^Tc.Ac^2 = \frac{1}{15}, \\ &b^Tc.A^2c = \frac{1}{30}, b^T(Ac)^2 = \frac{1}{20}, b^TAc^3 = \frac{1}{20}, \\ &b^T(Ac).(Ac) = \frac{1}{40}, b^TA^2c^2 = \frac{1}{60}, b^TA^3c = \frac{1}{120}, \end{aligned}$$

Plutôt que de chercher à satisfaire toutes ces équations directement, il est classique de faire les hypothèses suivantes, dites simplificatrices :

$$\begin{aligned} &\forall v \in \mathbb{R}^s,\ b^TAv = b^Tv - b^T(c.v), \\ &(Ac)_i = (\frac{c^2}{2})_i, i = 1,3,4,5,6, \\ &b_2 = 0. \end{aligned}$$

Il est alors aisé de vérifier que les 17 conditions d'ordre précédentes se réduisent maintenant aux 7 équations ci-dessous

$$\begin{aligned} b^T c^{k-1} &= \frac{1}{k}, k = 1, \dots, 5, \\ b^T c.A^2 c &= \frac{1}{30}, \\ b^T c.Ac^2 &= \frac{1}{15}, \end{aligned}$$

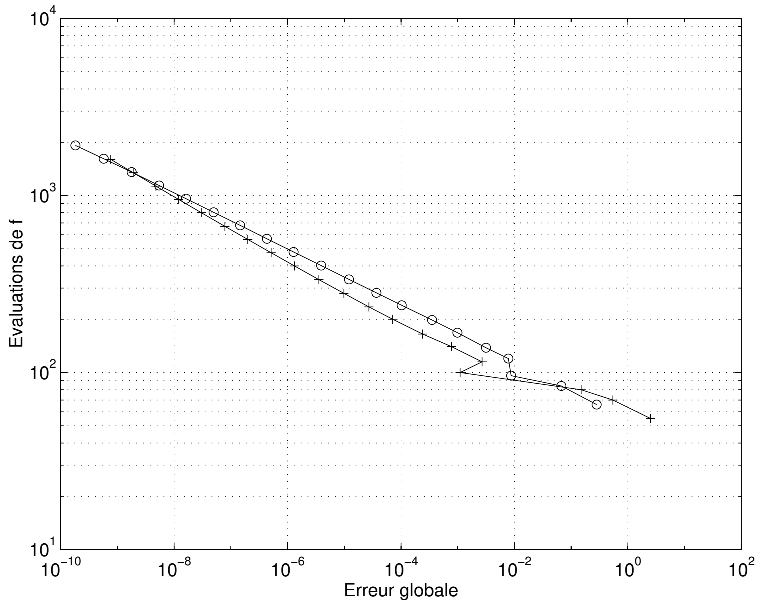
qui peuvent être résolues aisément. En imposant en outre que l'un des éléments sous-diagonaux de A soit nul, on obtient par exemple (pour un choix arbitraire des coefficients libres) la méthode «parallèle» suivante

$$\begin{array}{c|cccccc} 0 & & & & & & \\ \frac{1}{5} & 1/5 & & & & & \\ \frac{2}{5} & 0 & 2/5 & & & & \\ \frac{3}{5} & \frac{-3}{10} & \frac{9}{10} & 0 & & & \\ \frac{4}{5} & \frac{1}{5} & \frac{-2}{5} & 1 & 0 & & \\ 1 & \frac{-13}{2} & \frac{35}{2} & \frac{-15}{2} & -10 & \frac{15}{2} & \\ \hline & \frac{17}{144} & 0 & \frac{25}{36} & \frac{-25}{72} & \frac{25}{48} & \frac{1}{72} \end{array}.$$

A titre de comparaison, on a intégré ici l'équation de Van der Pol

$$\left\{ \begin{array}{ll} y_1' = y_2 & y_1(0) = 2.008619860874843136 \\ y_2' = (1 - y_1^2) y_2 - y_1 & y_2(0) = 0 \end{array} \right. , \; x \in [0, 6.6632868593231302], \quad (3.2)$$

avec la méthode construite ci-dessus d'une part et la méthode d'ordre 5 de Dormand et Prince d'autre part, considérée comme l'une des plus performantes disponibles à ce jour. Les résultats sont présentés sur le graphe ci-après, où le nombre d'évaluations séquentielles du second membre de (3.2) nécessaires à l'obtention d'une solution approchée au point $X = 6.6632868593231302$ est fonction de l'erreur globale au point X.



3.1.2 Méthodes PIRK («Parallel Iterated Runge-Kutta»)

Une manière beaucoup plus systématique de construire des méthodes parallèles optimales (i.e. avec $\sigma = p$), consiste à user de l'itération du point fixe. Étant donnée une méthode de Runge-Kutta $\mathcal{R} = (A, b, c)$ à s étapes, explicite ou implicite, on peut définir une nouvelle méthode par les équations suivantes

$$
\begin{aligned}
Y_i^{(0)} &= y_0, \ i = 1, \dots, s \\
Y_i^{(l)} &= y_0 + h \sum_{j=1}^{s} a_{i,j} f(x_0 + c_j h, Y_j^{(l-1)}), \ i = 1, \dots, s, \ l = 1, \dots, \sigma - 1, \\
y_1 &= y_0 + h \sum_{j=1}^{s} b_j f(x_0 + c_j h, Y_j^{(\sigma-1)}).
\end{aligned}
$$

Cet algorithme peut être interprété comme une méthode de Runge-Kutta explicite, de tableau

$$
\begin{array}{c|ccccc}
0 & 0 & & & & \\
c & A & 0 & & & \\
c & 0 & A & 0 & & \\
\vdots & \vdots & \ddots & \ddots & & \\
c & 0 & \dots & 0 & A & 0 \\
\hline
 & 0 & \dots & 0 & 0 & b^T
\end{array}
\quad . \tag{3.3}
$$

Si s processeurs sont disponibles, elle requiert σ évaluations de f séquentielles par pas. Pour déterminer l'ordre d'une telle méthode, on suppose une fois encore que f est lipschitzienne, de constante de Lipschitz L. Il vient alors,

$$
\max_i \|Y_i^{(l)} - Y_i\| \leq hLs \max_{i,j}(|a_{i,j}|) \max_i \|Y_i^{(l-1)} - Y_i\|,
$$

où les Y_i désignent les étapes internes de la méthode $\mathcal{R}$. On a donc le résultat suivant :

Théorème 8 *La méthode de Runge-Kutta parallèle (3.3) est d'ordre*

$$
\min(p_0, \sigma),
$$

si p_0 désigne l'ordre de la méthode de base.

En prenant $\sigma = p_0$, on obtient ainsi une méthode optimale.

3.2 Parallélisation des méthodes implicites

Comme cela a déjà été dit, les méthodes de Runge-Kutta implicites nécessitent à chaque pas la résolution d'un système à $s \times m$ équations de la forme

$$
Y_i = y_0 + h \sum_{j=1}^{s} a_{i,j} f(x_0 + c_j h, Y_j), i = 1, \dots, s. \tag{3.4}
$$

En règle générale, (3.4) doit être résolu de façon itérative. La méthode du point fixe est ici inadaptée, car elle détruit la stabilité, c'est-à-dire ce pour quoi il est nécessaire de recourir à une méthode implicite. On lui préférera donc la méthode de Newton. Rappelons que si $G(Y)$ est une fonction de $\mathbb{R}^n$ dans $\mathbb{R}^n$ différentiable, l'itération de Newton s'écrit

$$Y^{(k+1)} = Y^{(k)} - \left(\frac{\partial G}{\partial Y}\right)^{-1}_{Y=Y^{(k)}} G(Y^{(k)}). \tag{3.5}$$

Sous cette forme, chaque itération de (3.5) est encore extrêmement coûteuse puisqu'elle exige le calcul d'un jacobien et la résolution du système linéaire associé, c'est-à-dire essentiellement la décomposition LU (lower-upper) du jacobien. Dans la pratique, on se contente donc de calculer la matrice jacobienne au point $Y^{(0)}$. C'est cette matrice et sa décomposition LU qu'on utilise jusqu'à convergence de la méthode de Newton, connue dans cette variante sous le nom de méthode de Newton simplifiée. En résumé, la solution de (3.4) est obtenue par application successive de la formule[1]

$$(I - hA \otimes J)(Y^{(k+1)} - Y^{(k)}) = -Y^{(k)} + (e \otimes y_0) + h(A \otimes I)F(Y^{(k)}), \tag{3.6}$$

où $Y^{(k)} = ((Y_1^{(k)})^T, \dots, (Y_s^{(k)})^T)^T$, $F(Y^{(k)}) = \left(f(x_0 + c_1 h, Y_1^{(k)})^T, \dots, f(x_0 + c_s h, Y_s^{(k)})^T\right)^T$ et $J = \left(\frac{\partial f}{\partial y}\right)_{y=Y^{(0)}}$. Si m est la dimension du système différentiel, chaque évaluation de J exige $\mathcal{O}(m^2)$ opérations élémentaires, alors que la décomposition LU de $(I - hA \otimes J)$ nécessite $\mathcal{O}(s^3.m^3)$ opérations. C'est donc là que se concentre la plus grande partie du coût calcul, et c'est donc là que des gains importants peuvent être obtenus. Suivant la forme de A, diverses techniques permettent de diminuer le coût calcul par pas. Ainsi, si A est triangulaire inférieure, il est possible de découpler le système en s sous-systèmes. Si en outre les valeurs diagonales de A sont toutes égales à λ, une seule décomposition LU de la matrice $(I - h\lambda J)$ est nécessaire. De nombreuses méthodes basées sur cette idée et dénommées SDIRK pour «Singly Diagonally Implicit Runge-Kutta» ont été construites par R. Alexander et M. Crouzeix par exemple.

3.2.1 Méthodes MIRK (Multiply Implicit Runge-Kutta)

En raison de leur inadéquation aux problèmes raides ou algébro-différentiels, les méthodes DIRK ou SDIRK ont désormais été quelque peu abandonnées au profit des méthodes dites SIRK pour «Singly Implicit Runge-Kutta» ou MIRK. L'idée fondamentale consiste à écrire (3.5) dans un système de coordonnées appropriées : si $A = T^{-1}\Lambda T$, où Λ est diagonale ou diagonale par blocs, alors on peut prémultiplier (3.6) par $(hAT)^{-1} \otimes I$. Il vient alors

$$(\frac{1}{h}\Lambda \otimes I - I \otimes J)(T^{-1} \otimes I)(Y^{(k+1)} - Y^{(k)}) = -(\frac{1}{h}\Lambda \otimes I)(T^{-1} \otimes I)(Y^{(k)} - e \otimes y_0) + (T^{-1} \otimes I)F(Y^{(k)}).$$

En posant $W^{(k)} = (T^{-1} \otimes I)(Y^{(k)} - e \otimes y_0)$, l'itération devient donc

$$(\frac{1}{h}\Lambda \otimes I - I \otimes J)(W^{(k+1)} - W^{(k)}) = -(\frac{1}{h}\Lambda \otimes I)W^{(k)} + (T^{-1} \otimes I)F\left((T \otimes I)W^{(k)}\right), \tag{3.7}$$

1. Si $G \in \mathbb{R}^{u \times v}$ et $H \in \mathbb{R}^{w \times x}$, le produit tensoriel $G \otimes H$ de G et H est la matrice $uw \times vx$ par bloc

$$\begin{pmatrix} g_{11}H & g_{12}H & \dots & g_{1v}H \\ g_{21}H & g_{22}H & \dots & g_{2v}H \\ \vdots & & & \vdots \\ g_{u1}H & g_{u2}H & \dots & g_{uv}H \end{pmatrix}.$$

de sorte qu'une fois encore, il est possible de découpler le système en sous-systèmes. Si toutes les valeurs propres de A sont simples (cas des méthodes MIRK), alors on peut supposer Λ diagonale. Les s décompositions LU nécessaires peuvent alors être calculées en parallèle si l'on dispose de s processeurs.

Exemple : On reprend ici l'exemple du paragraphe 2.4.4, l'objectif étant maintenant d'obtenir une matrice A diagonalisable. Avec $s = 2$, la méthode de Gauss ne convient pas. On se contente donc désormais de l'ordre 3. On suppose en outre que $Ae = c$ et $Ac = \frac{1}{2}c^2$, et on pose $c^2 = \delta_1 e + \delta_2 c$, $V = [e, c]$. Il vient alors

$$V^{-1}AV = \begin{pmatrix} 0 & \frac{\delta_1}{2} \\ 1 & \frac{\delta_2}{2} \end{pmatrix}.$$

Les conditions $b^T e = 1$, $b^T c = 1/2$, et $b^T c^2 = 1/3$ impliquent d'autre part $\delta_2 = 2/3 - 2\delta_1$, d'où

$$P_A(x) = x^2 + (\delta_1 - 1/3)x - \frac{1}{2}\delta_1.$$

Il est alors facile de vérifier que P_A a ses racines réelles et positives ssi $\delta_1 \leq \frac{-2-\sqrt{3}}{3}$ ou $\frac{-2+\sqrt{3}}{3} \leq \delta_1 \leq 0$. Quant à la fonction de stabilité $R(z)$, on a

$$\begin{aligned} R(z) &= 1 + zb^T(I - zA)^{-1}e, \\ &= 1 + zb^T V(I - zV^{-1}AV)^{-1}V^{-1}e, \\ &= 1 + z[1, 1/2]\begin{pmatrix} 1 & -z\frac{\delta_1}{2} \\ -z & 1 - z\frac{\delta_2}{2} \end{pmatrix}^{-1} e_1. \end{aligned}$$

Donc $|R(z)| \leq 1$ sur l'axe des imaginaires, ssi $\delta_1 < -1/6$. La méthode est donc A-stable pour $\delta_1 \leq -1/6$. Pour $\delta_1 = -5/4$, on obtient par exemple

$$\begin{array}{c|cc} \frac{19}{12} - \frac{\sqrt{181}}{12} & -\frac{91\sqrt{181}}{4344} + \frac{19}{24} & -\frac{271\sqrt{181}}{4344} + \frac{19}{24} \\ \frac{19}{12} + \frac{\sqrt{181}}{12} & \frac{271\sqrt{181}}{4344} + \frac{19}{24} & \frac{91\sqrt{181}}{4344} + \frac{19}{24} \\ \hline & \frac{13\sqrt{181}}{362} + 1/2 & 1/2 - \frac{13\sqrt{181}}{362} \end{array}.$$

3.2.2 Méthodes PDIRK

Les méthodes PDIRK («Parallel Diagonal-implicitly Iterated Runge-Kutta») ont été introduites et étudiées par B. Sommeijer et P.J. Van der Houwen. L'idée est ici de remplacer le système implicite (3.5) par l'équation

$$Y^{(k)} - h(D \otimes I)F(Y^{(k)}) = e \otimes y_0 + h((A - D) \otimes I)F(Y^{(k-1)}),\ k = 1, \dots, \tag{3.8}$$

que l'on itère jusqu'à convergence, afin de conserver les propriétés intrinsèques de la méthode de base. A chaque itération, il faut encore résoudre un système implicite. Cependant, si D est choisie diagonale, alors il est possible de découpler le système et de le résoudre en parallèle. Pour que cette technique puisse être efficace, il est crucial que $Y^{(k)}$ tende rapidement vers la solution exacte Y de (3.5). On peut montrer qu'un facteur essentiel de la vitesse de convergence est la matrice $D^{-1}A - I$. Plus cette matrice est «proche» de 0, meilleure est la convergence.

Théorème 9 *L'itération (3.8) appliquée au problème test $y'(x) = \lambda y(x)$ est convergente pour $\lambda h \to \infty$ si et seulement si $\rho(D^{-1}A - I) < 1$.*

démonstration : pour $y' = \lambda y$, (3.8) devient plus simplement

$$Y^{(k)} - zDY^{(k)} = y_0 e + z(A - D)Y^{(k-1)}, \; z = \lambda h$$

c'est-à-dire

$$Y^{(k)} = y_0(I - zD)^{-1} e + z(I - zD)^{-1}(A - D)Y^{(k-1)}.$$

Pour $\Delta_k = Y^{(k)} - Y$, où Y est la solution exacte de

$$Y = y_0 e + zAY,$$

il vient

$$\Delta_k \;\; = \;\; z(I - zD)^{-1}(A - D)\Delta_{k-1}.$$

Pour $z \to \infty$, $\Delta_k = (D^{-1}A - I)\Delta_{k-1}$ de sorte que l'itération est convergente si et seulement si $\rho(D^{-1}A - I) < 1$. $\square$

Chapitre 4

Méthodes générales linéaires

Introduites par J.C. Butcher en 1966 [5], puis sous leur forme actuelle par K. Burrage et J. C. Butcher en 1980 [4] comme un moyen d'unifier la théorie des méthodes multipas et de Runge-Kutta, elles ont par la suite été étudiées en tant que telles. La classe des méthodes générales linéaires (MGLs) est suffisamment vaste pour inclure comme cas particuliers la quasi-totalité des méthodes classiques et d'autres méthodes plus exotiques apparues au cours des années.

4.1 Formalisme

Définition 8 *Une méthode générale linéaire est un schéma numérique qui, étant donnée une approximation initiale $y^{[0]}$ de la solution au point x_0, permet de calculer une approximation $y^{[N]}$ au point $x_N = x_0 + Nh$ par les équations de récurrence suivantes*

$$\left.\begin{array}{l} Y_i^{[n]} = h\sum_{j=1}^{s} a_{ij} f(x_n + c_j h, Y_j^{[n]}) + \sum_{j=1}^{r} u_{ij} y_j^{[n]},\ i = 1, \cdots, s \\ y_i^{[n+1]} = h\sum_{j=1}^{s} b_{ij} f(x_n + c_j h, Y_j^{[n]}) + \sum_{j=1}^{r} v_{ij} y_j^{[n]},\ i = 1, \cdots, r \end{array}\right\},\ n = 0, \cdots, N-1. \quad (4.1)$$

Le vecteur $y^{[n]}$ approche une fonction $\tilde{y}(x_n, h)$ dite «fonction valeur exacte»,qui fournit l'interprétation de la méthode. Les Y_i, $i = 1, \cdots, s$ sont dits «approximations internes» alors que les $y_j^{[n]}$, $j = 1, \cdots, r$ sont appelés «approximations externes».

Remarques :

- On a coutume de représenter une MGL $\mathcal{G} = (A, U, B, V)$ par son tableau

$$\begin{array}{c|c} A & U \\ \hline B & V \end{array}. \quad (4.2)$$

- Pour qu'une MGL soit d'un intérêt pratique non nul, on supposera toujours ici, qu'il est possible d'extraire de sa fonction valeur exacte $\tilde{y}(x_n, h)$ une approximation de la solution exacte $y(x_n)$.

- Les approximations internes Y_i sont calculées de manière transitoire. Elles ne sont utilisées qu'au sein du pas n.

- Les approximations externes contiennent par contre l'information nécessaire à la poursuite du processus d'intégration.

- Si le système différentiel (1.1) est de dimension m, alors chaque approximation, qu'elle soit interne ou externe, est de dimension m. En général, Y_i approche la solution exacte $y(x_n + c_i h)$, alors que $y_i^{[n]}$ approche une quantité $\tilde{y}_i(x_n, h)$ relative à la solution exacte. Il peut s'agir par exemple de $y(x_n)$, $hy'(x_n)$, ..., $h^k y^{(k)}(x_n)$ ou encore de combinaisons linéaires de ces différentes quantités.

Exemple : Une méthode de Runge-Kutta $\mathcal{R} = (A_{RK}, b_{RK}, c_{RK})$ à s_{RK} étapes peut s'écrire comme une MGL avec $r = 1$ et $s = s_{RK}$. Son tableau est de la forme

$$\begin{array}{c|c} & 1 \\ A_{RK} & \vdots \\ & 1 \\ \hline b_{RK}^T & 1 \end{array},$$

et sa fonction valeur exacte se réduit à $\tilde{y}(x, h) = y(x)$.

4.2 Stabilité et ordre d'approximation

L'étude de la stabilité, de l'ordre et de la convergence des méthodes générales linéaires est due pour l'essentiel à R. Skeel (1976).

Définition 9 *Une MGL est dite stable ou zéro-stable ssi*

$$\exists K \in \mathbb{R}, \forall n \in \mathbb{N}, \|V^n\| \leq K. \tag{4.3}$$

Il est clair qu'une MGL est stable si et seulement si

- toutes les valeurs propres de V sont à l'intérieur ou sur la frontière du disque unité,
- toutes les valeurs propres de module 1 sont non-défectives.

Par ailleurs, 1 est toujours valeur propre de V. En effet, lorsque h tend vers 0, $\tilde{y}(x, h)$ tend vers $(\alpha_0 \otimes y(x))$, α_0 étant un vecteur non nul de $\mathbb{R}^r$. En faisant tendre h vers 0 dans (4.1), il vient

$$(\alpha_0 \otimes y(x)) = (V \otimes I_m)(\alpha_0 \otimes y(x)),$$

ce qui implique que

$$V\alpha_0 = \alpha_0.$$

Exemple : Si $\tilde{y}(x, h)$ est le vecteur dit de Nordsieck,

$$\begin{bmatrix} y(x) \\ hy'(x) \\ \vdots \\ \frac{h^{r-1}}{(r-1)!} y^{(r-1)}(x) \end{bmatrix},$$

alors $\tilde{y}(x, h)$ tend vers $(y(x)^T, 0, \cdots, 0)^T$ lorsque h tend vers zéro. α_0 est donc ici le vecteur $(1, 0, \cdots, 0)^T \in \mathbb{R}^r$.

Pour une méthode stable, la décomposition canonique de Jordan de V est donc la matrice par bloc suivante

$$V = T \begin{bmatrix} \begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{pmatrix} & 0 & \cdots & \cdots & 0 \\ 0 & \begin{pmatrix} \xi_2 & & \\ & \ddots & \\ & & \xi_2 \end{pmatrix} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \begin{pmatrix} \xi_l & & \\ & \ddots & \\ & & \xi_l \end{pmatrix} & 0 \\ 0 & \cdots & \cdots & 0 & \tilde{J} \end{bmatrix} T^{-1},$$

où les $\xi_i, i = 2, \cdots, l$ désignent les valeurs propres de V de module 1 (hormis 1 elle-même), et $\tilde{J}$ la matrice par blocs correspondant aux valeurs propres de module strictement inférieur à 1. La projection spectrale E sur l'espace propre associé à la valeur propre 1 de V s'écrit alors $E = Tdiag(I, 0, \cdots, 0)T^{-1}$.

Définition 10 *Soit $\mathcal{G} = (A, U, B, V)$ une MGL de fonction valeur exacte $\tilde{y}(x, h)$. L'erreur locale $\delta^{[n+1]}$ est l'erreur d'approximation de $\mathcal{G}$ après un pas en partant de la solution exacte $\tilde{y}(x_n, h)$ au point x_n. Elle est définie par*

$$\delta^{[n+1]} = \tilde{y}(x_n + h, h) - y^{[n+1]}, \tag{4.4}$$

où

$$\begin{aligned} y^{[n+1]} &= h(B \otimes I_m)F(Y^{[n]}) + (V \otimes I_m)\tilde{y}(x_n, h), \\ Y^{[n]} &= h(A \otimes I_m)F(Y^{[n]}) + (U \otimes I_m)\tilde{y}(x_n, h). \end{aligned}$$

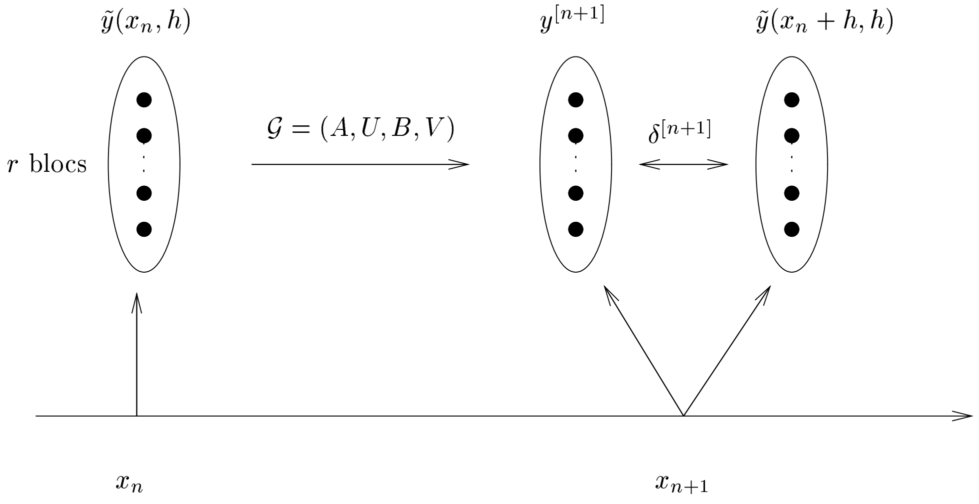


FIG. 4.1: *Définition de l'erreur locale*

Si f est suffisamment différentiable, alors il est possible de développer $\delta^{[n+1]}$ en série de Taylor au point x_n :

$$\delta^{[n+1]} = \delta_0(x_n) + h\delta_1(x_n) + \cdots + h^p\delta_p(x_n) + \mathcal{O}(h^{p+1}).$$

Définition 11 *Une méthode générale linéaire $\mathcal{G} = (A, U, B, V)$ est dite d'ordre p si et seulement si*

$$\forall n \in [0, N-1],\ \delta^{[n+1]} = \mathcal{O}(h^p),\ \text{et } \delta_p(x) = (v \otimes d(x)),\ Ev = 0. \tag{4.5}$$

Remarque : Cette définition fait apparaître le rôle déterminant de la projection E dans l'accumulation des erreurs. Notons que lorsque $r = 1$, -c'est le cas rappelons-le des méthodes de Runge-Kutta-, $E = 1$ et l'on retrouve la définition habituelle de l'ordre.

4.3 Convergence

Considérons maintenant simultanément les solutions numériques obtenues par la méthode $\mathcal{G}$

$$\begin{array}{ll} 1. & y^{[0]} = \phi(x_0, h), \\ 2. & n = 0, \cdots, N-1, \left\{ \begin{array}{lcl} y^{[n+1]} & = & h(B \otimes I_m)F(Y^{[n]}) + (V \otimes I_m)y^{[n]} \\ Y^{[n]} & = & h(A \otimes I_m)F(Y^{[n]}) + (U \otimes I_m)y^{[n]} \end{array} \right. , \end{array} \tag{4.6}$$

et par la méthode perturbée $\hat{\mathcal{G}}$

$$\begin{array}{ll} 1. & \hat{y}^{[0]} = \phi(x_0, h) + \epsilon_0, \\ 2. & n = 0, \cdots, N-1, \left\{ \begin{array}{lcl} \hat{y}^{[n+1]} & = & h(B \otimes I_m)F(\hat{Y}^{[n]}) + (V \otimes I_m)\hat{y}^{[n]} + \epsilon_{n+1} \\ \hat{Y}^{[n]} & = & h(A \otimes I_m)F(\hat{Y}^{[n]}) + (U \otimes I_m)\hat{y}^{[n]} \end{array} \right. . \end{array} \tag{4.7}$$

Notons $\Delta y^{[n]} = \hat{y}^{[n]} - y^{[n]}$, $\Delta Y^{[n]} = \hat{Y}^{[n]} - Y^{[n]}$ et $\Delta F^{[n]} = F(\hat{Y}^{[n]}) - F(Y^{[n]})$. Il vient :

$$\Delta y^{[n+1]} = (V \otimes I_m)\Delta y^{[n]} + h(B \otimes I_m)\Delta F^{[n]} + \epsilon_{n+1}.$$

Une récurrence facile conduit alors à

$$\Delta y^{[n+1]} = \sum_{k=0}^{n+1}(V \otimes I_m)^{n+1-k}\epsilon_k + h\sum_{k=0}^{n}(V \otimes I_m)^{n-k}(B \otimes I_m)\Delta F^{[k]}.$$

En supposant que $\mathcal{G}$ est stable, on obtient la majoration suivante [1]

$$\|\Delta y^{[n+1]}\| \leq \|\sum_{k=0}^{n+1}(V \otimes I_m)^{n+1-k}\epsilon_k\| + hK\sum_{k=0}^{n}\|(B \otimes I_m)\|\|\Delta F^{[k]}\|.$$

De plus, si f est Lipschitzienne de constante de Lipschitz L et h assez petit, on a

$$\|\Delta Y^{[n]}\| \leq \frac{1}{1 - hL\|A \otimes I_m\|}\|U \otimes I_m\|\|\Delta y^{[n]}\|,$$

et donc

$$\|\Delta F^{[n]}\| \leq \frac{L}{1 - hL\|A \otimes I_m\|}\|U \otimes I_m\|\|\Delta y^{[n]}\|.$$

1. On rappelle que $(V \otimes I_m)^k = (V^k \otimes I_m)$ pour tout k de $\mathbb{N}$.

En posant $K_0 = \frac{K L}{1-h_0 L\|A\otimes I_m\|}\|U\otimes I_m\|\|B\otimes I_m\|$ pour h_0 assez petit, on obtient

$$\forall h \leq h_0, \|\Delta y^{[n+1]}\| \leq \|\sum_{k=0}^{n+1}(V\otimes I_m)^{n+1-k}\epsilon_k\| + hK_0\sum_{k=0}^{n}\|\Delta y^{[k]}\|,$$

et une nouvelle récurrence donne finalement pour N suffisamment grand

$$\begin{aligned}\forall n \leq N, \|\Delta y^{[n]}\| &\leq (1+hK_0)^n \max_{n=0,\cdots,N}\left\|\sum_{k=0}^{n}(V\otimes I_m)^{n-k}\epsilon_k\right\|,\\ &\leq e^{(X-x_0)K_0}\max_{n=0,\cdots,N}\left\|\sum_{k=0}^{n}(V\otimes I_m)^{n-k}\epsilon_k\right\|.\end{aligned}$$

Lemme 2 *Soit $y^{[n]}$ la suite d'approximations (4.6) obtenue par une méthode générale linéaire $\mathcal{G} = (A, U, B, V)$ et soit $\hat{y}^{[n]}$ la suite d'approximations (4.7) obtenue en perturbant $\mathcal{G}$. Alors il existe une constante C telle que*

$$\max_{n=0,\cdots,N}\|\Delta y^{[n]}\| \leq C\max_{n=0,\cdots,N}\left\|\sum_{k=0}^{n}(V\otimes I_m)^{n-k}\epsilon_k\right\|. \tag{4.8}$$

Théorème 10 *Soit $\mathcal{G} = (A, U, B, V)$ une MGL stable et consistente d'ordre p. Alors $\mathcal{G}$ est convergente, d'ordre p, c'est-à-dire*

$$\exists C \in \mathbb{R}, \forall n \leq N, \|\tilde{y}(x_n, h) - y^{[n]}\| \leq Ch^p.$$

démonstration : bien que ce théorème soit vrai en toute généralité, on supposera ici que $\mathcal{G}$ est strictement stable, c'est-à-dire que 1 est la seule valeur propre de module 1. En prenant $\hat{y}^{[n]} = \tilde{y}(x_n, h)$ et $\epsilon_n = \delta^{[n]}$ dans le lemme précédent, on obtient

$$\max_{n=0,\cdots,N}\|\Delta y^{[n]}\| \leq C\max_{n=0,\cdots,N}\left\|\sum_{k=0}^{n}(V\otimes I_m)^{n-k}\delta^{[k]}\right\|. \tag{4.9}$$

Il suffit alors de majorer le second membre de l'égalité précédente. On considère pour cela la décomposition spectrale de V en $V = E + \tilde{V}$, $\tilde{V} = T diag(0,\cdots,0,\tilde{J})T^{-1}$. Il est aisé de voir que

$$V^k = E + \tilde{V}^k.$$

Il vient alors

$$\begin{aligned}\left\|\sum_{k=0}^{n}(V\otimes I_m)^{n-k}\delta^{[k]}\right\| &\leq h^p\left\|\sum_{k=0}^{n}(V\otimes I_m)^{n-k}\delta_p(x_k)\right\| + C.\sum_{k=0}^{n}h^{p+1},\\ &\leq h^p\left\|\sum_{k=0}^{n}Ev\otimes d(x_k) + \tilde{V}^{n-k}v\otimes d(x_k)\right\| + C\,(X-x_0)h^p,\\ &\leq h^p C_{EN}\left(\sum_{k=0}^{n}\|\tilde{V}\|^{n-k}\right)\max_{x\in[x_0,X]}\|d(x)\| + C\,(X-x_0)h^p.\end{aligned}$$

Or, $\rho(\tilde{V}) < 1$, donc il existe une norme telle que $\|\tilde{V}\| \leq \mu < 1$. Pour cette norme,

$$\sum_{k=0}^{n}\|\tilde{V}\|^{n-k} \leq \frac{1-\mu^{n+1}}{1-\mu}.$$

Le résultat est maintenant immédiat. □

Remarque : Dans le cas général, il faut recourir au lemme de sommation partielle d'Abel.

4.4 Étude de l'erreur locale

Le paragraphe précédent a permis d'établir des conditions suffisantes pour qu'une MGL soit convergente d'ordre p. Ces conditions font intervenir le développement en série de ce qui a été appelée erreur locale. On cherche donc désormais à calculer les coefficients de chacun des termes de ce développement.

Définition 12 *(E. Hairer et G. Wanner) Soit a une fonction de T dans $\mathbb{R}$. On appelle B-série, ou série de Butcher, la série*

$$B(a,y) = \sum_{t \in T} \frac{h^{\rho(t)}}{\rho(t)!} \alpha(t) a(t) F(t)(y). \tag{4.10}$$

Définition 13 *Un sous-arbre t^* de t est défini récursivement par (et noté $t^* \leq t$)*

1. $t^* = \emptyset$ *si* $t = \emptyset$.
2. $t^* = [t_1^*, \cdots, t_n^*]$ *si* $t = [t_1, \cdots, t_n]$.

On étend maintenant la fonction γ du chapitre 2 aux couples formés d'un arbre et d'un sous-arbre de cet arbre.

Définition 14 *La fonction γ est définie récursivement sur $\{(t^*, t) \in T \times T,\, t^* \leq T\}$ par*

1. $\gamma(\emptyset \leq t) = 1$.
2. $\gamma([t_1^*, \cdots, t_n^*] \leq [t_1, \cdots, t_n]) = \frac{\rho([t_1, \cdots, t_n])}{\rho([t_1^*, \cdots, t_n^*])} \prod_{i=1}^{n} \gamma(t_i^* \leq t_i)$.

Définition 15 *Soit $t \in T$ et soit $t^* \leq t$. On appelle ensemble différence (et on note $d_{t^*}(t)$) l'ensemble des arbres obtenus lorsque t^* est extrait de t.*

Le théorème fondamental suivant permet de calculer les coefficients de la composée de deux B-séries. Il permettra le calcul effectif des conditions d'ordre pour les méthodes générales linéaires. Notons qu'il permet également de démontrer le théorème 5 du chapitre 2.

Théorème 11 *Soient a et b deux fonctions de T dans $\mathbb{R}$. Supposons que $a(\emptyset) = 1$. Alors la composée $B(b, B(a,y))$ est encore une B-série $B(ab, y)$ et ses coefficients $ab(t)$ sont donnés par*

$$\forall t \in T,\, ab(t) = \sum_{t^* \leq t} \gamma(t^* \leq t)\, b(t^*) \prod_{s \in d_{t^*}(t)} a(s).$$

Si l'on suppose maintenant que $\tilde{y}(x,h)$ possède un développement en série de Butcher $\tilde{y}(x,h) = B(\nu, y(x))$ [2], alors l'erreur locale est elle-même développable en B-série :

$$\delta^{[n+1]} = B(\delta, y(x)).$$

D'après (4.1), on a alors

$$\begin{aligned} B(\delta, y(x)) &= B(\nu, y(x+h)) - (B \otimes I_m) B(Y', y(x)) - (V \otimes I_m) B(\nu, y(x)), \\ B(Y, y(x)) &= (A \otimes I_m) B(Y', y(x)) + (U \otimes I_m) B(\nu, y(x)), \end{aligned}$$

2. Ici, ν est une application de T dans $\mathbb{R}^r$ et donc par $B(\nu, y(x))$ on entend $\sum_{t \in T} \frac{h^{\rho(t)}}{\rho(t)!} \alpha(t) \left(\nu(t) \otimes F(t)(y)(x)\right)$.

où $Y = B(Y, y(x))$ et $hF(B(Y, y(x))) = B(Y', y(x))$ [3]. En appliquant une première fois le théorème 11 avec $a(.) = Y(.)$ et $b(\tau) = 1$, $b(t \neq \tau) = 0$, il vient

$$\begin{aligned} Y'(\emptyset) &= 1, \\ Y'([t_1, \cdots, t_n]) &= \rho([t_1, \cdots, t_n]) \prod_{i=1}^{n} a(t_i). \end{aligned}$$

D'autre part, si l'on désigne par $S\nu$ les coefficients de $B(S\nu, y(x)) = B(\nu, y(x+h))$, le théorème 11 avec $a(.) = 1$ et $b(.) = \nu(.)$ donne

$$S\nu(t) = \sum_{t^* \leq t} \gamma(t^* \leq t)\nu(t^*).$$

Si l'on suppose en outre que $\nu(t^*)$ ne dépend que de l'ordre de t^*, alors [4]

$$S\nu(t) = \sum_{k=0}^{\rho(t)} \begin{pmatrix} \rho(t) \\ k \end{pmatrix} \nu(\tau^k).$$

Théorème 12 *Soit $\mathcal{G} = (A, U, B, V)$ une méthode générale linéaire de fonction valeur exacte $\tilde{y}(x, h) = B(\nu, y(x))$. L'erreur locale de $\mathcal{G}$ est la B-série de coefficients*

$$\begin{aligned} \forall\, t = [t_1, \cdots, t_n] \in T,\ \delta(t) &= \sum_{t^* \leq t} \gamma(t^* \leq t)\nu(t^*) - V\nu(t) - \rho(t) B \prod_{i=1}^{n} Y(t_i), \\ Y(t) &= U\nu(t) + \rho(t) A \prod_{i=1}^{n} Y(t_i). \end{aligned}$$

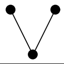
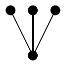
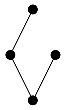
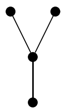
$\mathcal{G}$ est d'ordre p si et seulement si

$$\left\{ \begin{array}{l} \forall\, t \in T,\ \rho(t) \leq p-1,\ \delta(t) = 0 \\ \forall\, t \in T,\ \rho(t) = p,\ E\delta(t) = 0 \end{array} \right. .$$

Les conditions jusqu'à l'ordre 4 sont données dans le tableau suivant sous les hypothèses [5]

$$\begin{array}{ll} (H1) & \nu(\emptyset) - V\nu(\emptyset) = 0, \\ (H2) & U\nu(\tau) + Ae = c. \end{array}$$

3. Y et $Y' = hf(Y)$ désignent ici à la fois les approximations internes et leurs dérivées, mais aussi les applications de T dans $\mathbb{R}^s$ leur correspondant.
4. On note $\tau^k = [\underbrace{\tau, \cdots, \tau}_{k-1}]$.
5. (H1) et (H2) impliquent que $Y^{[n]} = (y(x_n + c_1 h)^T, \cdots, y(x_n + c_s h)^T)^T + \mathcal{O}(h^2)$.

Ordre	Graphe	Condition d'ordre
1	$t_1 =$ 	$\nu(\emptyset) + \nu(t_1) - V\nu(t_1) - Be = 0.$
2	$t_2 =$ 	$\nu(\emptyset) + 2\nu(t_1) + \nu(t_2) - V\nu(t_2) - 2Bc = 0$
3	$t_{31} =$ 	$\nu(\emptyset) + 3\nu(t_1) + 3\nu(t_2) + \nu(t_{31}) - V\nu(t_{31}) - 3Bc^2 = 0$
3	$t_{32} =$ 	$\nu(\emptyset) + 3\nu(t_1) + 3\nu(t_2) + \nu(t_{32}) - V\nu(t_{32})$ $-3BU\nu(t_2) - 6BAc = 0$
4	$t_{41} =$ 	$\nu(\emptyset) + 4\nu(t_1) + 6\nu(t_2) + 4\nu(t_{31}) + \nu(t_{41}) - V\nu(t_{41})$ $-4Bc^3 = 0$
4	$t_{42} =$ 	$\nu(\emptyset) + 4\nu(t_1) + 6\nu(t_2) + \frac{8}{3}\nu(t_{31}) + \frac{4}{3}\nu(t_{32}) + \nu(t_{42}) - V\nu(t_{42})$ $-4B(c.(U\nu(t_2) + 2Ac)) = 0$
4	$t_{43} =$ 	$\nu(\emptyset) + 4\nu(t_1) + 6\nu(t_2) + 4\nu(t_{32}) + \nu(t_{43}) - V\nu(t_{43})$ $-4BU\nu(t_{31}) - 12BAc^2 = 0$
4	$t_{44} =$ 	$\nu(\emptyset) + 4\nu(t_1) + 6\nu(t_2) + 4\nu(t_{32}) + \nu(t_{44}) - V\nu(t_{44}) -$ $4BU\nu(t_{32}) - 12BAU\nu(t_2) - 24BA^2c = 0$

4.5 Construction d'une méthode générale linéaire d'ordre 3

On se place dans le cas $r = s = 2$ et on suppose que V est de rang 1, ce qui assure la zéro-stabilité de la méthode. On choisit en outre $\tilde{y}(x,h)$ telle que $\tilde{y}_1(x,h) = y(x) + \mathcal{O}(h^3)$ et $\tilde{y}_2(x,h) = hy'(x) + \mathcal{O}(h^3)$. On a donc

$$\nu(\emptyset) = (1,0)^T,$$
$$\nu(\tau) = (0,1)^T,$$
$$\nu([\tau]) = (0,0)^T.$$

Les conditions d'ordre s'écrivent alors aisément. En projetant suivant E[6] les conditions portant sur $\delta([[\tau]])$ et $\delta([\tau,\tau])$, on s'aperçoit que $\nu([[\tau]])$ et $\nu(([\tau,\tau])$ n'interviennent pas. On obtient alors par exemple

$$\begin{bmatrix} A & U \\ B & V \end{bmatrix} = \left[\begin{array}{cc|cc} \frac{4}{9} & -\frac{1}{9} & 1 & \frac{1}{6} \\ \frac{11}{18} & \frac{2}{9} & 1 & \frac{1}{6} \\ \hline \frac{2}{3} & \frac{1}{6} & 1 & \frac{1}{6} \\ 0 & 1 & 0 & 0 \end{array}\right], \; c = [1/2, 1]^T.$$

6. Ici, $E = \nu(\emptyset)v^T$, $v = (1, v_2)^T$.

Chapitre 5

Diagonally Implicit Multi Stage Integration Methods

Dans quatre articles publiés récemment, -J.C. Butcher [7, 8, 9] et J.C. Butcher et Z. Jackiewicz [10], J.C. Butcher définit une nouvelle classe de méthodes générales linéaires. Cette classe, désignée par l'acronyme DIMSIM (pour «Diagonally Implicit Multi Stage Integration Method»), peut elle-même être divisée en sous-classes suivant la nature du problème (raide ou non-raide) et la nature de la méthode (séquentielle ou parallèle). La forme générique des méthodes DIMSIM est celle des méthodes générales linéaires, c'est-à-dire

$$\forall i \in [1,s], Y_i^{[n]} = h\sum_{j=1}^{s} a_{i,j} f(x_n + c_j h, Y_j^{[n]}) + \sum_{j=1}^{r} u_{i,j} y_j^{[n]}, \tag{5.1}$$

$$\forall i \in [1,r], y_i^{[n+1]} = h\sum_{j=1}^{s} b_{i,j} f(x_n + c_j h, Y_j^{[n]}) + \sum_{j=1}^{r} v_{i,j} y_j^{[n]}, \tag{5.2}$$

mais avec une matrice A de structure particulière :

$$A = \begin{pmatrix} \lambda & 0 & \dots & 0 \\ a_{21} & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ a_{s1} & a_{s2} & \dots & \lambda \end{pmatrix}. \tag{5.3}$$

L'idée sous-jacente des méthodes DIMSIM est d'obtenir des méthodes conjuguant la précision et la stabilité des méthodes de Runge-Kutta sans le coût prohibitif des méthodes totalement implicites. Les quatre sous-classes correspondent à différents choix de λ et des coefficients sous-diagonaux de A (voir tableau 5.1). Lorsque $\lambda = 0$, la méthode est explicite. Elle est implicite dans les autres cas. Ici, nous nous restreindrons à l'étude des méthodes parallèles, c'est-à-dire de types 3 et 4, suivant que le problème est raide ou non.

Type	1	2	3	4
$A_{i,j}$	Arbitraire	Arbitraire	0	0
λ	0	$\neq 0$	0	$\neq 0$
Problème	Non-raide	Raide	Non-raide	Raide
Parallélisme	Limité	Limité	Intrinsèque	Intrinsèque

TAB. 5.1: «*Diagonally Implicit Multi Stage Integration Methods*» *de types 1 à 4*

5.1 Conditions d'ordre pour les méthodes DIMSIM

On supposera ici, -et c'est une autre caractéristique fondamentale des méthodes DIMSIM-, que la fonction valeur exacte est de la forme

$$\forall i \in [1, r],\ \tilde{y}_i(x_n, h) = \sum_{j=0}^{p} \alpha_{ij} h^j y^{(j)}(x_0) \tag{5.4}$$

où les α_{ij}, $i = 1, \dots, r$, $j = 0, 1, \dots, p$ sont des paramètres réels caractérisant la méthode. On supposera en outre que les $Y_i^{[n]}$ vérifient les équations suivantes

$$Y_i^{[n]} = \sum_{j=0}^{q} \frac{c_i^j}{j!} h^j y^{(j)}(x_n) + \mathcal{O}(h^{q+1}), \tag{5.5}$$

ce qui signifie que les vecteurs $Y_i^{[n]}$ approchent la solution exacte $y(x)$ aux points $x_n + c_i h$ à l'ordre q. q est généralement appelé ordre des approximations internes. Les conditions d'ordre sont maintenant obtenues en imposant que $y^{[n+1]}$ ait le même développement en série de Taylor que $y^{[n]}$ jusqu'à l'ordre p, mais au point $x_{n+1} = x_n + h$:

$$\forall i \in [1, r],\ y_i^{[n+1]} = \sum_{j=0}^{p} \alpha_{i,j} h^j y^{(j)}(x_{n+1}) + \mathcal{O}(h^{p+1}). \tag{5.6}$$

On considère alors les cas suivants :

- $p = q$ (voir [7])
- $p = q + 1$ (voir [10])

Le cas le plus délicat correspond à $p = q + 1$: on distingue alors les sous-cas $s = k = q$ et $s - 1 = k - 1 = q$ (voir [10]). Soit donc

$$e^{cz} = [e^{c_1 z}, \dots, e^{c_s z}]^T, \tag{5.7}$$
$$\alpha_j = [\alpha_{1j}, \dots, \alpha_{rj}]^T, \tag{5.8}$$
$$w(z) = \sum_{j=0}^{p} \alpha_j z^j. \tag{5.9}$$

Les conditions d'ordre dans les différents cas sont regroupées dans les théorèmes suivant :

Théorème 13 *Soit $\mathcal{G} = (A, U, B, V)$ une méthode DIMSIM.*

- *Si $p = q$ (J.C. Butcher), $\mathcal{G}$ est d'ordre p ssi*

$$\left\{ \begin{array}{l} e^{cz} = zAe^{cz} + Uw(z) + \mathcal{O}(z^{p+1}) \\ e^{cz} w(z) = zBe^{cz} + Vw(z) + \mathcal{O}(z^{p+1}) \end{array} \right. . \tag{5.10}$$

- *Si $p = q + 1$ (J.C. Butcher and Z. Jackiewicz), $\mathcal{G}$ est d'ordre p ssi*

$$\left\{ \begin{array}{l} e^{cz} = zAe^{cz} + Uw(z) + (\frac{1}{p!}c^p - \frac{1}{(p-1)!}Ac^{p-1} - U\alpha_p)z^p + \mathcal{O}(z^{p+1}) \\ e^{z} w(z) = zBe^{cz} + Vw(z) + \mathcal{O}(z^{p+1}) \end{array} \right. . \tag{5.11}$$

démonstration : définissons tout d'abord les matrices

$$W = [\alpha_0, \cdots, \alpha_p] \in \mathbb{R}^{r\times(p+1)}, X = [e, c, \frac{c^2}{2}, \cdots, \frac{c^p}{p!}] \in \mathbb{R}^{s\times(p+1)},$$

$$S = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & & & 1 & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{(p+1)\times(p+1)}, X' = [0, e, c, \frac{c^2}{2}, \cdots, \frac{c^{p-1}}{(p-1)!}] = XS \in \mathbb{R}^{s\times(p+1)},$$

et les vecteurs $\tilde{N}(x,h) = (y(x)^T, \cdots, h^p y^{(p)}(x)^T)^T$ et $Z = (1, z, \cdots, z^p)$. L'ordre des approximations internes étant q, on a $Y_i = y(x_n + c_i h) + \mathcal{O}(h^{q+1})$, $i = 1, \cdots, s$. Donc

$$(X \otimes I)\tilde{N}(x_n, h) = (A \otimes I)(X' \otimes I)\tilde{N}(x_n, h) + (U \otimes I)(W \otimes I)\tilde{N}(x_n, h) + \mathcal{O}(h^{q+1}).$$

soit en regroupant

$$((X - AX' - UW) \otimes I)\tilde{N}(x_n, h) = \mathcal{O}(h^{q+1}).$$

Si les e_i désignent les vecteurs de la base canonique de $\mathbb{R}^p$, alors on a

$$\forall i \in \{1, \cdots, q\}, \ (X - AX' - UW)e_i = 0,$$

c'est-à-dire encore

$$\begin{aligned} (X - AX' - UW)Z = 0, &\quad \text{si p=q,} \\ (X - AX' - UW)Z - (\frac{c^p}{p!} - A\frac{c^{p-1}}{(p-1)!} - U\alpha_p)z^p = 0, &\quad \text{si p=q+1.} \end{aligned}$$

Le résultat pour les approximations internes résulte alors des équations

$$e^{cz} = XZ + \mathcal{O}(z^{p+1}) \text{ et } ze^{cz} = X'Z + \mathcal{O}(z^{p+1}).$$

En ce qui concerne les approximations externes, il vient

$$\tilde{N}(x_n + h, h) = (BX' \otimes I)\tilde{N}(x_n, h) + (VW \otimes I)\tilde{N}(x_n, h) + \mathcal{O}(h^{p+1}),$$

soit

$$\left((\sum_{k=0}^{p} \frac{1}{k!} WS^k - BX' - VW) \otimes I \right) \tilde{N}(x_n, h) = \mathcal{O}(h^{p+1}).$$

Et on obtient comme précédemment

$$(\sum_{k=0}^{p} \frac{1}{k!} WS^k - BX' - VW)Z = 0.$$

Le résultat est alors une conséquence immédiate des équations suivantes

$$BX'Z = zBe^{cz} + \mathcal{O}(z^{p+1}) \text{ et } \sum_{k=0}^{p} \frac{1}{k!} WS^k Z = e^z WZ + \mathcal{O}(z^{p+1}) = e^z w(z) + \mathcal{O}(z^{p+1}).$$

□.

Soient B_0, B_1 et B_2 les matrices définies par

$$B_0 = \left(\frac{\int_0^{1+c_i} \Phi_j(x)dx}{\Phi_j(c_j)} \right)_{i=1,s;j=1,r}, \tag{5.12}$$

$$B_1 = \left(\frac{\Phi_j(1+c_i)}{\Phi_j(c_j)} \right)_{i=1,s;j=1,r}, \tag{5.13}$$

$$B_2 = \left(\frac{\int_0^{c_i} \Phi_j(x)dx}{\Phi_j(c_j)} \right)_{i=1,s;j=1,r}, \tag{5.14}$$

avec $\Phi_j(x) = \prod_{l=1,l\neq j}^{s}(x - c_l)$.

Théorème 14 *(J.C. Butcher and Z. Jackiewicz) Soit $\mathcal{G} = (A, U, B, V)$ une méthode DIMSIM telle que $s = r$ et $U = I$.*

- *Si $p = q + 1 = s + 1 = r + 1$, $\mathcal{G}$ est d'ordre p et d'ordre interne q ssi*

$$\begin{cases} B = B_0 - AB_1 - VB_2 + VA \\ Bc^{p-1} = \frac{1}{p}((e+c)^p - c^p) - A((e+c)^{p-1} - c^{p-1}) - (p-1)!(V-I)\alpha_p. \end{cases} \tag{5.15}$$

- *Si $p = q + 1 = s = r$, $\mathcal{G}$ est d'ordre p et d'ordre interne q ssi*

$$B = B_0 - AB_1 - VB_2 + VA + (V - I)Q \tag{5.16}$$

avec $Q = (\frac{1}{p}c^p - Ac^{p-1} - (p-1)!\alpha_p)e^T T^{-1}$, $T = diag[\Phi_1(c_1), \Phi_2(c_2), \ldots, \Phi_s(c_s)]$.

Le théorème suivant permet la construction effective de méthodes telles que $p = q + 1$. Il fournit en effet une condition nécessaire et suffisante pour qu'une méthode d'ordre q et d'ordre interne q soit d'ordre $q + 1$.

Théorème 15 *(J.C. Butcher and Z. Jackiewicz) Soit $\mathcal{G} = (A, U, B, V)$ une méthode DIMSIM d'ordre q telle que 1 soit valeur propre simple de V, et telle que l'ordre des approximations internes soit q. Si e^z est une valeur propre approchée de $M(z) = V + zB(I - zA)^{-1}U$ d'ordre $q+1$, c'est-à-dire telle que*

$$det(e^z I - M(z)) = \mathcal{O}(z^{q+2}) \tag{5.17}$$

alors $\mathcal{G}$ est d'ordre $p = q + 1$.

5.2 Exemples de méthodes DIMSIM de types 3 et 4

Ce paragraphe démontre l'intérêt des théorèmes précédents sur deux exemples simples.

Exemple : Cet exemple est emprunté à [10]. Il illustre le théorème 15 dans le cas $q = k = s = 2$ et $p = 3$. De la structure des matrices

$$A = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}, U = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, V = \begin{pmatrix} v & 1-v \\ v & 1-v \end{pmatrix}, \tag{5.18}$$

et de l'équation (5.15), on tire

$$B = \begin{pmatrix} (\frac{1}{2}+\lambda)v & (\frac{1}{2}-\lambda)v \\ (-\frac{1}{2}+\lambda) + (\frac{1}{2}+\lambda)v & (\frac{1}{2}-\lambda)v + \frac{3}{2} - \lambda \end{pmatrix}. \tag{5.19}$$

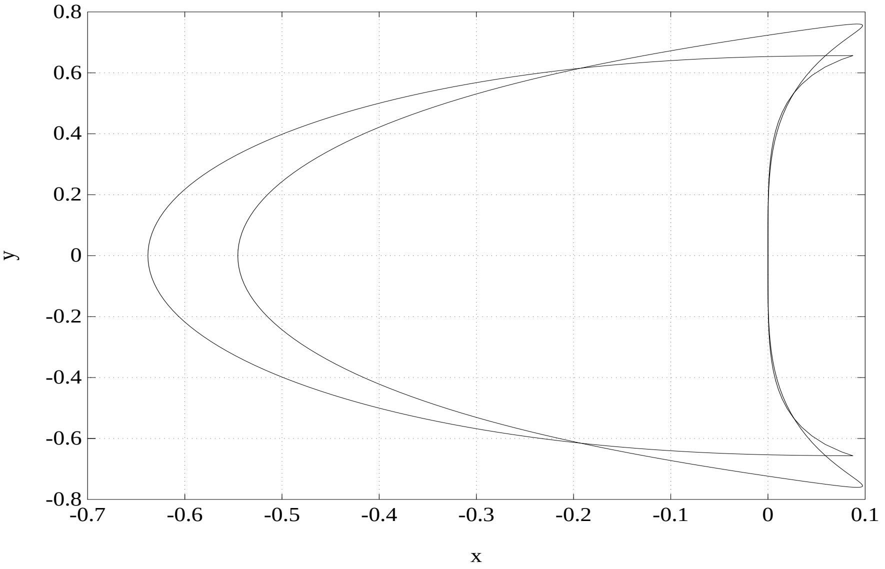


FIG. 5.1: *Domaines de stabilité ($z = x + iy$) comparés de la méthode ci-avant et de la méthode d'Adams d'ordre 3.*

La condition (5.17) s'écrit alors

$$det(e^z I - M(z)) = (\frac{5}{12} - \lambda - \frac{1}{2}v + \lambda v)z^3 + \mathcal{O}(z^4) = \mathcal{O}(z^4) \tag{5.20}$$

c'est-à-dire

$$v = \frac{5 - 12\lambda}{6(1 - 2\lambda)}, \quad \lambda \neq \frac{1}{2}. \tag{5.21}$$

En prenant $\lambda = 0$ (type 3) on obtient finalement B et V :

$$B = \begin{pmatrix} \frac{5}{12} & \frac{5}{12} \\ \frac{-1}{12} & \frac{23}{12} \end{pmatrix}, \quad V = \begin{pmatrix} \frac{5}{6} & \frac{1}{6} \\ \frac{5}{6} & \frac{1}{6} \end{pmatrix}. \tag{5.22}$$

Si l'on dispose d'un ordinateur multi-processeur, cette méthode a un coût calcul par pas équivalent à celui d'une méthode multipas explicite classique. Son domaine de stabilité est néanmoins légèrement plus étendu que celui de la méthode d'Adams d'ordre 3 (voir figure 5.1). C'est l'un des avantages possibles des méthodes DIMSIM.[1]

Dans [7], J.C. Butcher utilise une technique plus systématique, basée sur la transformation mentionnée précédemment. Elle conduit alors à une famille complète de méthodes explicites de type 3. Le vecteur c y est choisi de façon à construire des méthodes proches des méthodes de Runge-Kutta ($c_i \in [0, 1]$) ou au contraire proches des méthodes multipas ($c = (-(r-2), \cdots, 0, 1)^T$).

Exemple : Si l'on prend cette fois-ci dans 5.2 $\lambda \neq 0$ (type 4), alors le polynôme de stabilité de la méthode s'écrit

$$\Phi(w, z) = (1 - \lambda z)^2 w^2 - p_1(z)w + p_2(z) \tag{5.23}$$

1. Pour une comparaison plus juste de deux méthodes, il convient également de prendre en compte les constantes d'erreur.

avec

$$p_1(z) = 1 + \frac{7 - 24\lambda + 18\lambda^2}{3(1-2\lambda)} z - \frac{7\lambda - 21\lambda^2 + 12\lambda^3}{3(1-2\lambda)} z^2, \tag{5.24}$$

$$p_2(z) = \frac{4 - 12\lambda + 6\lambda^2}{3(1-2\lambda)} z + \frac{5 - 20\lambda + 24\lambda^2 - 12\lambda^3}{6(1-2\lambda)} z^2. \tag{5.25}$$

L'étude de ce polynôme montre alors que la méthode est A-stable ssi $\lambda \geq \lambda_0$ avec $\lambda_0 \approx 0.76985$. De sorte que pour ces valeurs de λ, la méthode obtenue est plus stable qu'une méthode BDF d'ordre 3 tout en ayant le même coût calcul.

Chapitre 6

Parallélisme à travers les pas

Afin d'illustrer la notion de parallélisme à travers les pas, on étudie dans un premier temps la parallélisation des systèmes linéaires de la forme

$$\begin{cases} y'(x) = A(x)y(x) + g(x) \\ y(x_0) = y_0, \quad x \in [x_0, X] \end{cases} \tag{6.1}$$

La solution exacte de (6.1) est donnée par (voir par exemple [13], pp. 64)

$$y(x) = R(x, x_0)y_0 + \int_{x_0}^{x} R(x, s)g(s)ds, \tag{6.2}$$

où $R(x, x_0)$ est la matrice résolvante et est définie par le système

$$\begin{cases} \frac{dR(x,x_0)}{dx} = A(x)R(x, x_0) \\ R(x_0, x_0) = I \end{cases} \tag{6.3}$$

En tenant compte de l'équation $R(x, x_0) = R(x, x')R(x', x_0)$, on peut encore écrire la solution (6.2) au point X sous la forme

$$y(X) = R(X, x)\left(R(x, x_0)y_0 + \int_{x_0}^{x} R(x, s)g(s)ds\right) + \int_{x}^{X} R(X, s)g(s)ds \tag{6.4}$$

et plus généralement, en se basant sur les points de la subdivision $x_0, x_1, \ldots, x_N = X$ de $[x_0, X]$,

$$y(X) = \sum_{n=0}^{N-1} R(X, x_{n+1}) \int_{x_n}^{x_{n+1}} R(x_{n+1}, s)g(s)ds + R(X, x_0)y_0. \tag{6.5}$$

Il est aisé de voir que $z_{n+1} = \int_{x_n}^{x_{n+1}} R(x_{n+1}, s)g(s)ds$ est également la solution au point x_{n+1} du système linéaire

$$\begin{cases} z'(x) = A(x)z(x) + g(x) \\ z(x_n) = 0, \quad x \in [x_n, x_{n+1}] \end{cases} \tag{6.6}$$

En notant z_0 la valeur initiale y_0, on obtient finalement pour (6.2)

$$y(X) = \sum_{n=0}^{N} R(X, x_n)z_n = z_N + \sum_{n=0}^{N-1} \left(\prod_{l=0}^{N-1-n} R(x_{n+l+1}, x_{n+l}) \right) z_n. \tag{6.7}$$

Si N peut s'écrire comme une puissance de 2 ($N = 2^K$), et si l'on dispose de N processeurs, alors $y(X)$ peut être calculé au moyen d'une réduction cyclique en $\mathcal{O}(K)$ étapes. Les détails de l'algorithme de réduction cyclique sont donnés par exemple dans [12]. Il est donc possible dans la situation très simple des problèmes linéaires de subdiviser l'intervalle d'intégration en N sous-intervalles et de calculer la contribution de chaque intervalle indépendemment de celle des autres. La solution au point final est alors obtenue en «assemblant» les résultats partiels au moyen d'une récurrence qui peut elle-même parallélisée. Donc, si l'on suppose que la majeure partie des calculs se concentre sur la détermination des z_n, c'est-à-dire l'intégration du système sur les intervalles $[x_n, x_{n+1}]$, alors l'algorithme ainsi défini est intrinsèquement parallèle. Il serait cependant abusif de négliger les difficultés d'implantation : en particulier, la multiplication des matrices résolvantes devient rapidement impossible pour de grands systèmes. Il reste que l'idée mise en évidence ici et qui est à l'origine des méthodes exposées dans ce chapître s'apparente à la décomposition de domaine «en temps».

6.1 Méthodes «waveform relaxation»

L'idée de base des méthodes «waveform relaxation» apparaît naturellement lorsque l'on considère la formulation intégrale du problème différentiel (1.1) et l'itération dite de Picard associée :

$$\left\{ \begin{array}{l} y^{(0)}(x) \equiv y_0 \\ i = 0, \dots, \quad (y^{(i+1)})'(x)) = f(x, y^{(i)}(x)) \end{array} \right. \tag{6.8}$$

ou encore de façon équivalente

$$\left\{ \begin{array}{l} y^{(0)}(x) = y_0 \\ i = 0, \dots, \quad y^{(i+1)}(x) = \int_{x_0}^{X} f(t, y^{(i)}(t))dt. \end{array} \right. \tag{6.9}$$

A partir d'une solution approchée $y^{(i)}$ de la solution exacte $y(x)$ sur tout l'intervalle d'intégration $[x_0, X]$, une nouvelle approximation $y^{(i+1)}$ est obtenue à l'itération $i+1$. Le second membre de (6.9) étant indépendant de y, chaque itération est en fait un problème de quadrature de la forme

$$y'(x) = f(x) \tag{6.10}$$

et peut donc être parallélisée : si N processeurs sont disponibles, chacun d'eux peut calculer $f(x_n)$ en parallèle et $y(x_N)$ est alors obtenu par la formule de quadrature

$$y(x_N) = y(x_0) + \sum_{n=0}^{N} \omega_n f(x_n) \tag{6.11}$$

en $\mathcal{O}(log(N))$ étapes. Le point crucial du procédé est donc la vitesse de convergence de l'itération. Dans l'espace des fonctions continues dans $[x_0, X]$, cette itération converge de façon superlinéaire si f est Lipschitzienne de constante de Lipschitz L. On a en effet,

$$\forall x \in [x_0, X], \quad \|y^{(1)}(x) - y(x)\| \leq L(x - x_0)\|y^{(0)} - y\|, \tag{6.12}$$

$$\forall x \in [x_0, X], \quad \|y^{(2)}(x) - y(x)\| \leq L^2 \frac{(x - x_0)^2}{2!}\|y^{(0)} - y\|, \tag{6.13}$$

$$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \tag{6.14}$$

et donc par récurrence sur i,

$$\forall x \in [x_0, X], \quad \|y^{(i)}(x) - y(x)\| \leq L^i \frac{(x - x_0)^i}{i!}\|y^{(0)} - y\|. \tag{6.15}$$

Il reste que la convergence peut être très lente sur un long intervalle. On est ainsi amené à considérer diverses techniques d'accélération de l'itération de Picard. Comme il a été noté par R. Skeel in [20], la majorité des procédés itératifs continus peuvent être obtenus à partir d'une décomposition de la fonction f. Ainsi, si $\mathcal{F}(x, y, z)$ désigne une fonction approchant $f(x, y)$ et vérifiant

$$\mathcal{F}(x, y, y) = f(x, y), \tag{6.16}$$

alors on peut remplacer (6.9) par

$$\begin{cases} y^{(0)}(x) \equiv y_0 \\ i = 0, \dots, \quad (y^{(i+1)})'(x) = \mathcal{F}(x, y^{(i)}(x), y^{(i+1)}(x)). \end{cases} \tag{6.17}$$

Il est clair qu'une représentation continue de la fonction $y^{(i)}$ sur l'intervalle $[x_0, X]$ est nécessaire. Cette représentation est ici appelée «waveform». En pratique, $y^{(i)}$ n'est véritablement nécessaire que sur une fenêtre $[x_b, x_e]$ qui se «déplace» le long de l'intervalle d'intégration $[x_0, X]$. Dans le cas des méthodes *waveform relaxation* (WR), $\mathcal{F}$ est de la forme

$$\mathcal{F}(x, y, z) = g(x, z) + f(x, y) - g(x, y). \tag{6.18}$$

Plusieurs choix sont possibles pour g (voir par exemple [12]). Cependant, seules les fonctions g dont le jacobien est diagonal ou diagonal par blocs autorisent le découplage des différentes composantes de y en vue d'une résolution parallèle. Un choix particulièrement attrayant semble être la fonction suivante :

$$g(x, z) = D(y)z \tag{6.19}$$

où $A(y) = N(y) + D(y)$ est le jacobien de f et $D(y)$ la sous-matrice diagonale où diagonale par blocs de $A(y)$.

Notons en outre, que quelle que soit la forme du jacobien de g, il est toujours possible d'avoir recours à la technique décrite dans l'introduction de ce paragraphe. Par exemple, la décomposition de Newton-Raphson conduit à une itération de la forme

$$\begin{cases} y^{(0)}(x) \equiv y_0 \\ i = 0, \dots, \quad (y^{(i+1)})'(x) = \frac{\partial f}{\partial y}(x, y^{(i)}(x))(y^{(i+1)}(x) - y^{(i)}(x)) + f(x, y^{(i)}(x)) \end{cases} \tag{6.20}$$

c'est-à-dire un système linéaire. Il reste alors à appliquer la méthode de parallélisation correspondante.

La convergence des méthodes WR est généralement assurée (voir par exemple [21]).Par contre, le choix de la décomposition de f permettant d'optimiser la vitesse de convergence reste indéterminé. O. Nevanlinna and U. Miekkala (voir [16, 17, 18, 19]) ont néanmoins analysé les relations entre la vitesse de convergence de l'itération et la forme de $\mathcal{F}$ dans le cas des problèmes linéaires. La complexité de cette analyse dans un cas considéré par ailleurs comme élémentaire laisse mal augurer de la situation générale.

6.2 Méthode de A. Bellen et M. Zennaro

Dans l'article [3], A. Bellen et M. Zennaro introduisent une classe d'algorithmes parallèles pour les problèmes de valeur initiale, applicables aux équations aux différences et donc, par extension, aux équations différentielles ordinaires. Ces algorithmes offrent des gains potentiels de vitesse d'exécution considérables. L'intégration numérique de (1.1) sur la subdivision $x_0, x_1, \ldots, x_N = X$ de $[x_0, X]$ par une méthode à un pas donne lieu à une équation aux différences de la forme

$$y_{n+1} = F_{n+1}(y_n) \tag{6.21}$$

où y_n approche la solution exacte au point x_n de la subdivision et où les fonctions F_n dépendent du schéma numérique sous-jacent. Remarquons que F_{n+1} peut éventuellement être définie implicitement si le schéma numérique employé est lui-même implicite.

L'idée principale de A. Bellen et M. Zennaro consiste à considérer (6.21) comme une fonction définie sur un ensemble de suites et à considérer la résolution de (1.1) comme un problème de point fixe. Plus précisément, si U désigne la suite $\{y_0, y_1, \ldots, y_n, \ldots\}$ et Φ la fonction

$$\Phi(U) = \{y_0, F_1(y_0), F_2(y_1), \ldots, F_{n+1}(y_n), \ldots\}, \tag{6.22}$$

alors, résoudre(1.1) revient à déterminer un point-fixe de Φ dans l'espace des suites à valeur dans $\mathbb{R}^m$. Afin de calculer ce point fixe de manière efficace, les auteurs utilisent une variante de la méthode de Newton, appelée méthode de Steffensen. Le lecteur intéressé pourra consulter l'article [3] pour plus de détails. L'itération est ici décrite sommairement :

1. Pour $n = 0, \ldots, N-1$, calculer $F_{n+1}(y_n^{(i)})$ et $A_{n+1} \approx \left.\frac{\partial F_{n+1}}{\partial y}\right|_{y=y_n^{(i)}}$.
2. Dérouler la récurrence :

$$y_{n+1}^{(i+1)} = F_{n+1}(y_n^{(i)}) + A_{n+1}(y_n^{(i+1)} - y_n^{(i)}). \tag{6.23}$$

L'avantage principal de cette formulation est son potentiel pour le parallélisme. Au sein de chaque itétation de la méthode de Steffensen, il est en effet possible de calculer les F_n et A_n en parallèle. En conséquence, chaque itération recquiert, sur une machine parallèle disposant de suffisamment de processeurs, un temps équivalent à celui d'une évaluation de F et de son jacobien, temps auquel il convient néanmoins d'ajouter celui nécessaire à l'exécution du processus de correction (6.23). Ainsi, si k^* iterations sont nécessaires, alors on peut espérer une accélération de N/k^*.

Comme dans le cas des méthodes «waveform relaxation», le point clé est donc la vitesse de convergence de l'itération, si convergence il y a! Dans [3] est démontré le résultat suivant :

Théorème 16 *L'itération de Steffensen appliquée au système (6.22) est localement quadratiquement convergente.*

Comme de coutume pour les méthodes apparentées à la méthode de Newton, la taille du «voisinage» où l'on est assuré de la convergence dépend des dérivées de la fonction dont on cherche un zéro, et donc ici des dérivées de Φ et des F_n. Malheureusement, ce voisinage est fréquemment très petit : à moins de connaître le comportement «qualitatif» de la solution exacte, l'approximation initiale (généralement prise égale à $y \equiv y_0$) a de fortes chances de se situer hors du voisinage de convergence.

Cependant, A. Bellen et M. Zennaro font état de résultats encourageants, démontrant que la convergence peut en fait être très rapide. On peut donner à cela deux explications : d'une part, il est toujours possible de réduire le pas ; puisque les fonctions F_n s'approchent de l'identité à mesure que le pas tend vers zéro, les dérivées de F_n peuvent être rendues très petites, autorisant par la une convergence plus rapide. D'autre part, la taille de la suite elle-même peut être réduite jusqu'à ce que l'approximation initiale se trouve dans le voisinage de convergence.

Bibliographie

[1] ARNOL'D, *Ordinary differential equations.*

[2] A. BELLEN, R. VERMIGLIO, AND M. ZENNARO, *Parallel ODE-solvers with step-size control*, Journal of Computational and Applied Mathematics, 31 (1990), pp. 277–293.

[3] A. BELLEN AND M. ZENNARO, *Parallel Algorithms for Initial Value Problems*, Journal of Computational and Applied Mathematics, 25 (1989), pp. 341–350.

[4] K. BURRAGE AND J. BUTCHER, *Non-linear stability of a general class of differential equation methods*, BIT, 20 (1993), pp. 185–203.

[5] J. BUTCHER, *On the convergence of numerical solutions to ordinary differential equations..*, Mathematics of Computation, 20 (1966), pp. 1–10.

[6] J. BUTCHER, *The numerical analysis of ordinary differential equations. Runge-Kutta and general linear methods.*, John Wiley and Sons, 1987.

[7] ——, *Diagonally Implicit Multi Stage Integration Methods*, Applied Numerical Mathematics, 11 (1993), pp. 347–363.

[8] ——, *A Transformation for the Analysis of DIMSIMs*, BIT, (1993).

[9] J. BUTCHER AND P. CHARTIER, *Parallel general linear methods for stiff ordinary differential and differential algebraic equations*, Applied Numerical Mathematics, 17 (1995), pp. 213–222.

[10] J. BUTCHER AND Z. JACKIEWICZ, *Diagonally implicit general linear methods for ordinary differential equations*, BIT, 33 (1993), pp. 452–472. (in preparation).

[11] M. CROUZEIX AND A. MIGNOT, *Analyse numérique des équations différentielles*, Masson, 1989.

[12] C. GEAR, *Parallel methods for ordinary differential equations*, Research R-87-1369, University of Illinois, Urbana, IL, 1986.

[13] E. HAIRER, S. NØRSETT, AND G. WANNER, *Solving Ordinary Differential Equations I. Nonstiff Problems*, vol. 1, Springer-Verlag, 1987.

[14] E. HAIRER AND G. WANNER, *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems.*, vol. 2, Springer-Verlag, 1991.

[15] A. ISERLES AND S. NØRSETT, *Order Stars*, Chapman and Hall, 1991.

[16] U. MIEKKALA AND O. NEVANLINNA, *Convergence of dynamic iteration methods for initial value problem*, SIAM Journal of Scientific and Statistical Computing, 8 (1987).

[17] ——, *Sets of convergence and stability regions*, BIT, 27 (1987), pp. 554–584.

[18] O. Nevanlinna, *Remarks on Picard-Lindelöf iteration, Part 1*, BIT, 29 (1989), pp. 328–346.

[19] ——, *Remarks on Picard-Lindelöf iteration, Part 2*, BIT, 29 (1989), pp. 535–562.

[20] R. Skeel, *Waveform Iteration and the Shifted Picard Splitting*, SIAM Journal of Scientific and Statistical Computing, 10 (1989), pp. 756–776.

[21] J. White, F. Odeh, A. Vincentelli, and A. Ruehli, *Waveform relaxation : theory and practice.*, Tech. Rep. UCB/ERL M85/65, UC Berkeley, 1985.