Intégration numérique des équations différentielles raides

Ernst Hairer et Gerhard Wanner
Dept. de Mathématiques, Université de Genève,
CH-1211 Genève 4, Switzerland,
e-mail: Ernst.Hairer@math.unige.ch, Gerhard.Wanner@math.unige.ch

Table des matières

1 Reconnaître une équation différentielle raide **3**
1.1 Méthodes standards pour équations différentielles 3
1.2 Phénomènes numériques avec les problèmes raides 5
1.3 Autres exemples de problèmes raides 6
1.4 Méthodes numériques simples pour équations raides 8
1.5 Définition et discussion de la A-stabilité 9

2 Intégrateurs numériques d'ordre supérieur **13**
2.1 Méthodes de Runge–Kutta implicites 13
2.1.1 Méthodes de collocation 13
2.1.2 Méthodes de Runge–Kutta diagonalement implicites 16
2.2 Méthodes multipas (BDF) 17
2.3 Méthodes d'extrapolation 19
2.4 Programmation des intégrateurs numériques 20

3 Méthodes pour des problèmes particuliers **22**
3.1 Méthodes explicites avec longues régions de stabilité 22
3.1.1 Méthodes de Runge–Kutta–Chebyshev 22
3.1.2 Approche de Lebedev, Medovikov et Abdulle (code ROCK) . . 23
3.2 Méthodes de séparation (splitting) 24
3.3 Méthodes implicites-explicites (IMEX) 26
3.3.1 Méthodes implicites-explicites de Runge–Kutta 26
3.3.2 Méthodes multipas implicites-explicites 28
3.4 Equations différentielles-algébriques 28

4 Appendice : Programmes informatiques **30**

Résumé

Dans de nombreuses applications, la dynamique d'un système peut être modélisée par des équations différentielles. L'étude de systèmes mécaniques (par exemple en astronomie ou en dynamique moléculaire), l'analyse des circuits électriques, ou la théorie du contrôle (robotique) nous fournissent de tels problèmes. Souvent, pour les problèmes dits raides, les méthodes standards ne fournissent pas une solution correcte en un temps de calcul acceptable.

Cet article récapitulatif explique les phénomènes qui apparaissent dans les équations différentielles raides, en s'appuyant sur des exemples issus des réactions chimiques ainsi que des équations aux dérivées partielles discrétisées en espace. Les propriétés essentielles des intégrateurs numériques pour la résolution des équations raides sont discutées (A-stabilité, domaine de stabilité). Pour des problèmes généraux les méthodes de Runge–Kutta implicites, les méthodes multipas (BDF) et les méthodes d'extrapolation sont traitées. Pour des problèmes raides particuliers de grande dimension, sont également abordées les méthodes explicites avec grande région de stabilité, les méthodes de séparation et les méthodes implicites-explicites. Une liste de programmes informatiques du domaine publique conclue l'article.

Comme références sur la résolution numérique des équations différentielles raides, on peut consulter les livres [Wil74], [Mir81], [DV84], [Aik85], [HW96] et [HV03].

1 Reconnaître une équation différentielle raide

Les équations différentielles **raides**, écrites sous forme autonome, sont des systèmes d'équations différentielles

$$\dot{y} = f(y) \qquad y(t_0) = y_0 \ ,$$

où les méthodes standards (explicites) se revèlent inefficaces. Il est cependant utile de commencer par présenter ces méthodes, car elles permettent de comprendre les phénomènes et les difficultés qui interviennent dans la résolution des problèmes raides.

1.1 Méthodes standards pour équations différentielles

Méthode d'Euler. L'algorithme le plus simple pour calculer des approximations numériques y_n de la solution $y(t)$ aux points t_1, t_2, $t_3, \ldots$ est donné par

$$y_{n+1} = y_n + hf(y_n) \qquad \text{où} \qquad h = t_{n+1} - t_n$$

est appelé **longueur de pas**. Cette formule représente le début de la série de Taylor pour la solution, et possède donc une erreur locale de $\mathcal{O}(h^2)$ pour $h \to 0$. L'erreur accumulée sur un intervalle fixe de longueur t, c.-à-d., après $\mathcal{O}(\frac{1}{h})$ pas de calcul, se comporte donc comme $\mathcal{O}(h)$, et nous disons que la méthode est **d'ordre 1**.

Méthode du trapèze. Une possibilité pour augmenter l'ordre est de rendre la formule symétrique

$$y_{n+1} = y_n + \frac{h}{2}\Big(f(y_n) + f(y_{n+1})\Big).$$

Cette formule possède une erreur locale de $\mathcal{O}(h^3)$, donc une erreur accumulée de $\mathcal{O}(h^2)$ et la méthode est dite d'ordre 2. Mais cela est au prix d'être **implicite**, c.-à-d., le calcul de y_{n+1} nécessite la résolution d'un système (en général non linéaire) d'équations algébriques.

Méthode de Runge. Pour transformer la méthode précédente en une méthode explicite, tout en conservant son ordre, nous utilisons l'idée du **prédicteur-correcteur**, c.-à-d., nous "prédisons" la valeur de y_{n+1} par un pas de la méthode d'Euler, que nous "corrigeons" ensuite par la formule du trapèze, ce qui donne

$$Y_1 = y_n, \qquad Y_2 = y_n + hf(Y_1), \qquad y_{n+1} = y_n + \frac{h}{2}\Big(f(Y_1) + f(Y_2)\Big).$$

Méthode de Kutta. En généralisant la classe de ces méthodes, après un développement en série de Taylor de l'erreur locale (voir les exemples dans le §2.1.2 ci-après), W. Kutta trouve la méthode d'ordre 4

$$Y_1 = y_n, \qquad Y_2 = y_n + \frac{h}{2}f(Y_1), \qquad Y_3 = y_n + \frac{h}{2}f(Y_2), \qquad Y_4 = y_n + hf(Y_3),$$
$$y_{n+1} = y_n + \frac{h}{6}\Big(f(Y_1) + 2f(Y_2) + 2f(Y_3) + f(Y_4)\Big) \ .$$

que nous écrivons sous la forme

$$Y_i = y_n + h\sum_{j=1}^{i-1} a_{ij} f(Y_j), \quad (i = 1, \dots, s)$$
$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i f(Y_i)$$

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

Cette méthode, qui généralise la règle de Simpson, fut pendant des décennies la méthode standard pour la résolution des équations différentielles.

Contrôle de la longueur des pas. Une importante amélioration des algorithmes fut l'introduction des méthodes avec contrôle de la longueur de pas h, dans les années 60. L'idée est de rajouter à la formule précédente une deuxième formule

$$\widehat{y}_{n+1} = y_n + h\sum_{i=1}^{s} \widehat{b}_i f(Y_i)$$

(normalement d'un ordre inférieur) et d'utiliser $err = \|y_{n+1} - \widehat{y}_{n+1}\|$ ou $err = \|y_{n+1} - \widehat{y}_{n+1}\|/\|y_{n+1}\|$ et le critère $err \leq Tol$ pour accepter ou rejeter un pas. Une formule de la forme

$$h_{new} = 0.9 \cdot h \left(\frac{Tol}{err}\right)^{1/r}$$

aide à trouver un meilleur h pour le pas suivant. On parle alors de méthodes emboitées. Les meilleures méthodes de ce type, connues actuellement, sont la méthode DOPRI5 d'ordre 5(4), de Dormand and Prince,

DOPRI5

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

et la méthode DOP853 d'ordre 8, dont la construction des coefficients est expliquée dans le livre [HNW93, §II.5].

Méthodes multipas (Adams). Une autre possibilité pour augmenter l'ordre des méthodes d'Euler et du trapèze est d'utiliser les valeurs $f_j = f(y_j)$ sur plusieurs pas antérieurs. Cela donne les méthodes **explicites** d'Adams (voir [HNW93, §III.1])

$$y_{n+1} = y_n + h\sum_{j=0}^{k-1} \gamma_j \nabla^j f_n \qquad \text{avec} \qquad \gamma_j = (-1)^j \int_0^1 \binom{-s}{j} ds$$

d'ordre k, et les méthodes **implicites** d'Adams

$$y_{n+1} = y_n + h \sum_{j=0}^{k} \gamma_j^* \nabla^j f_{n+1} \qquad \text{avec} \qquad \gamma_j^* = (-1)^j \int_0^1 \binom{-s+1}{j} ds$$

d'ordre $k+1$. Rappelons que $\nabla f_n = f_n - f_{n-1}$, et récursivement $\nabla^{j+1} f_n = \nabla^j f_n - \nabla^j f_{n-1}$. L'implémentation de ces méthodes dans des codes comme DEABM, VODE, ou LSODE est expliquée dans [HNW93, §III.7]. Nous reviendrons sur la théorie des méthodes multipas pour des problèmes raides dans le §2.2 ci-après.

1.2 Phénomènes numériques avec les problèmes raides

Le premier prototype d'une équation différentielle raide est apparu dans des calculs de réactions chimiques [CH52]. Il s'agit de problèmes où des réactions rapides se tiennent en équilibre pour produire des mouvements lents. Curtiss and Hirschfelder s'inspirent du modèle (voir Fig. 1)

$$\dot{y} = \lambda(y - \phi(t))$$

où $\phi(t)$ (pour notre exemple $\phi(t) = \arctan(20\,t)$) est une fonction donnée et λ (pour notre exemple $\lambda = -4,\ -20,\ -100$) est une constante négative de grande taille. Cette constante mesure la vitesse avec laquelle les solutions voisines se rapprochent (étape appelée **transition de phase**) vers une solution lente proche de $\phi(t)$. La raideur augmente en Fig. 1 de gauche à droite.

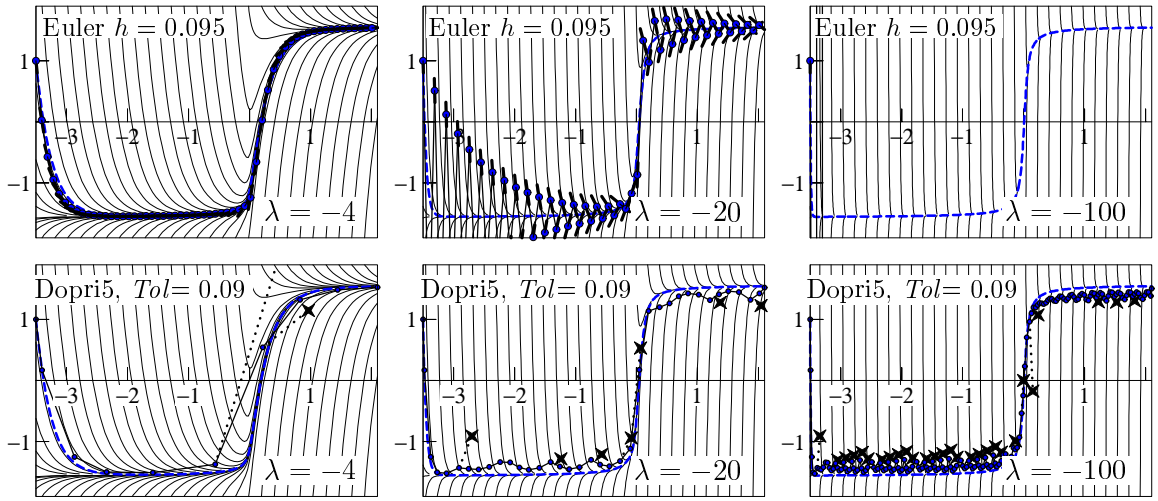

FIGURE 1: Équation de Curtiss–Hirschfelder $\dot{y} = \lambda(y - \arctan 20t)$; valeur initiale $y(-3.5) = 1$. Images du haut : méthode explicite à pas h fixe ; images du bas : code DOPRI5 avec contrôle du pas ; pas rejetés symbolisés par des croix noires.

Nous appliquons à ce problème une méthode explicite (Euler) avec longueur de pas h fixe ($h = 0.095$), et constatons que dès que la raideur dépasse un certain seuil, la solution numérique explose (troisième des images supérieurs en Fig. 1). Par contre,

un algorithme avec contrôle automatique des pas h (images inférieures) empêche cette explosion par la réduction automatique de h. Lorsque la raideur augmente, la longueur des pas et les pas rejetés (croix noires) ne dépendent plus de l'allure de la solution (comme dans la première image proche de $t = 0$), mais sont limités uniquement par la raideur du problème, indépendamment de la précision *Tol* prescrite. Plus la raideur devient forte, plus ces algorithmes deviennent inefficaces. Bien que l'œil humain voit immédiatement qu'il y a un problème avec cette solution en observant la Fig. 1, il est plus difficile de remarquer la raideur dans un calcul automatique. Néanmoins, des algorithmes pour la détection de la raideur ont été développés (voir [HW96, p. 21]). Ainsi, si la valeur de $-\lambda$ augmente encore d'avantage, le code DOPRI5 affiche le diagnostique "the problem appears to be stiff". Ce code possède aussi un dispositif pour "stabiliser" le contrôle de h, dans le but d'éviter ce grand nombre de pas rejetés. Ici, cette option a été supprimée pour notre démonstration.

1.3 Autres exemples de problèmes raides

Souvent, la raideur d'un problème saute aux yeux par la présence de coefficients de grande taille dans l'équation, comme pour l'équation de Curtiss–Hirschfelder ou la réaction de Robertson ([HW96, p. 3]).

L'Oregonator. Par contre, les équations du célèbre Oregonator, une réaction chimique entre $HBrO_2$, Br^-, et $Ce\,(IV)$ (voir [FN74] et [HNW93, p. 118])

$$\begin{aligned}
\dot{y}_1 &= 77.27\,(y_2 + y_1(1 - 8.375 \times 10^{-6} y_1 - y_2)) \\
\dot{y}_2 &= \frac{1}{77.27}\,(y_3 - (1 + y_1)y_2) \\
\dot{y}_3 &= 0.161\,(y_1 - y_3)
\end{aligned}$$

n'ont, à première vue, rien de méchant. Après l'état initial, que nous choisissons arbitrairemment comme $y_1(0) = 3$, $y_2(0) = 1$, $y_3(0) = 2$, la valeur de y_1 explose en premier, à cause du terme $\dot{y}_1 = 77.27\,y_1$, et y_2 tend vers zéro à cause de $\dot{y}_2 = -\frac{1}{77.27}\,y_1 \cdot y_2$. Ensuite y_3 explose avec $\dot{y}_3 = 0.161 y_1$, suivi par y_2 qui explose à son tour avec $\dot{y}_2 = \frac{1}{77.27} y_3$. Une fois que y_1 et y_2 atteignent une certaine taille, les termes négatifs dans l'équation pour $\dot{y}_1$ deviennent dominants et y_1 chute vers zéro. Tout cela se passe dans un intervalle de temps $0 \leq t \leq 3$ (voir Fig. 2). Ainsi commence la longue agonie de y_1 avec $\dot{y}_1 \approx -77.27 \cdot y_2 \cdot y_1$. C'est en fait ici que se cache la forte raideur du problème, car le facteur négatif -77.27 est multiplié par y_2, qui est de l'ordre de grandeur 10^3. Parce qu'on désire calculer la solution sur un intervalle de temps $3 \leq t \leq 300$, il faut encore rajouter un autre facteur 10^2 et la raideur du problème correspond finalement à une valeur de $\lambda \approx -10^2 \cdot 10^3 \cdot 10^2 = -10^7$. La solution s'approche d'un cycle limite et son comportement se repète périodiquement.

Il est sans espoir d'essayer de résoudre cette équation par une méthode explicite. Nous utilisons donc un des algorithmes les plus efficaces que nous ayons pour les

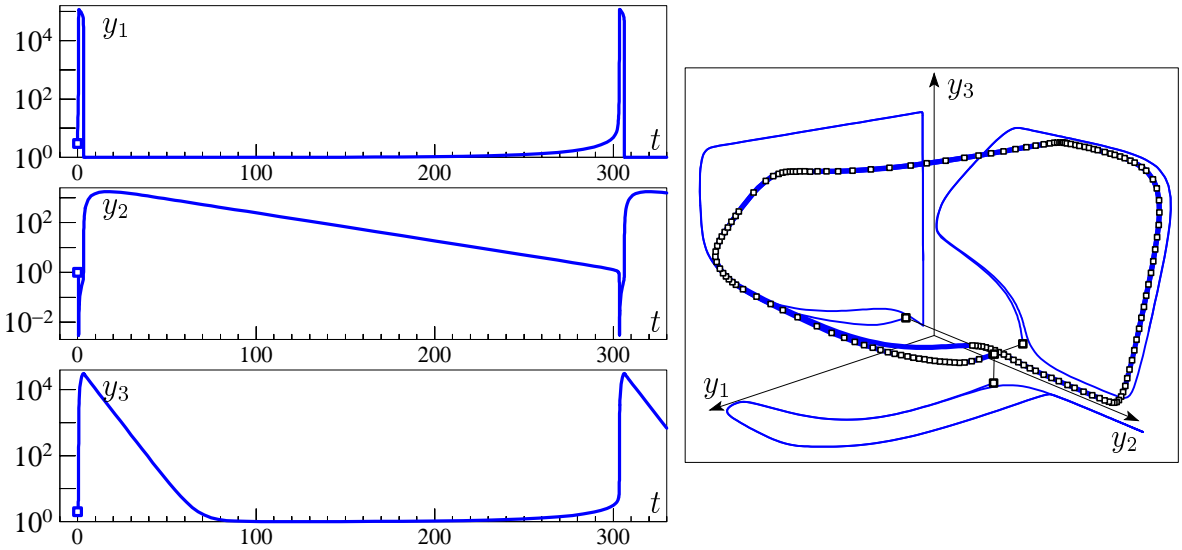

FIGURE 2: Solutions de l'Oregonator. À gauche comme fonction de t, à droite vision tridimensionelle du cycle limite, et (en petits carrés) les points calculés par RADAU5 avec $Tol = 10^{-4}$.

problèmes raides, le code RADAU5 (décrit au §2.1.1; voir aussi §4) qui calcule avec une précision de 10^{-4} une période de la solution en environ 150 points. Ces points sont indiqués dans la Fig. 2 à droite. La longueur h des pas, qui varie dans un intervalle entre 10^{-3} et 20, montre l'importance du contrôle automatique de h pour cet exemple.

Diffusion et problèmes paraboliques. Une autre source de problèmes raides est la discrétisation de la diffusion dans les problèmes paraboliques à l'aide de la méthode des éléments finis ou des différences finies en espace, écrite sous forme d'un large système d'équations différentielles en t. Nous choisissons comme exemple le problème appelé le **Brusselateur**, auquel nous rajoutons de la diffusion en dimension 1 d'espace $0 \leq x \leq 1$:

$$\dot{u}_i = \alpha \cdot \delta^2 u_i + 1 + u_i^2 v_i - 4u_i$$

$$\dot{v}_i = \alpha \cdot \delta^2 v_i + 3u_i - u_i^2 v_i$$

où $\delta^2 u_i = (u_{i+1} - 2u_i + u_{i-1})/\Delta x^2$, avec $\Delta x = 1/N$, représente la diffusion. Dans cet exemple, la réaction chimique, qui produit un cycle limite, n'est pas raide du tout, contrairement à la réaction d'Oregonator. Ici, c'est en fait la diffusion qui est raide. Les valeurs propres de la matrice tridiagonale $\mathrm{tridiag}(1, -2, 1)$ sont toutes réelles et varient entre -4 et 0. Ainsi, la raideur du problème correspond, pour $\alpha = 1/50$ et $N = 41$ (donc $1/\Delta x^2 = 1681$), à un $\lambda \approx -4\alpha/\Delta x^2 \approx -130$. Le problème est moyennement raide, et DOPRI5 reste encore applicable (voir Fig. 3 à gauche), mais un code spécialisé pour problèmes à valeurs propres réelles négatives comme ROCK4 (voir la description en §3.1.2 ci-dessous) est bien meilleur.

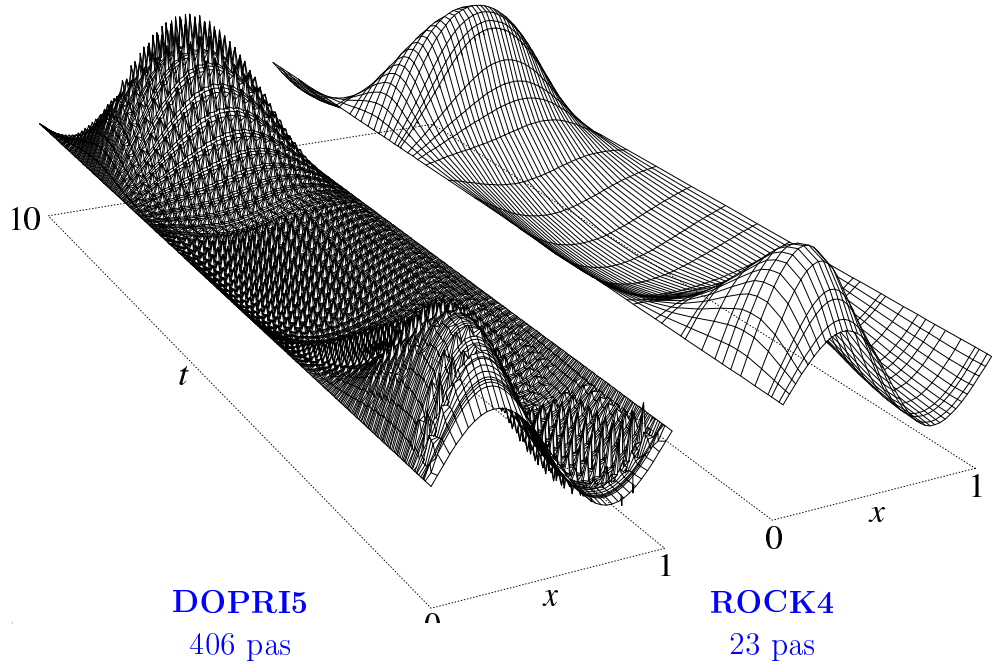

FIGURE 3: Solutions du Brusselateur avec diffusion. À gauche DOPRI5, à droite les solutions calculées par ROCK4. Valeurs initiales $u_i(0) = 1+\sin(2\pi x_i)$, $v_i(0) = 3$, conditions aux bords $u_0(t) = u_N(t) = 1$, $v_0(t) = v_N(t) = 3$, calcul pour l'intervalle de temps $0 \le t \le 10$.

1.4 Méthodes numériques simples pour équations raides

La règle du trapèze. Cette méthode, que nous avons chassée dans le §1.1, n'aimant pas son caractère implicite, revient maintenant nous secourir avec nos problèmes raides. Dans le cas de l'équation de Curtiss–Hirschfelder (§1.2), nous pouvons facilement résoudre la formule du trapèze pour obtenir

$$y_{n+1} = \Big(\Big(1 + \frac{\lambda h}{2}\Big) y_n - \frac{\lambda h}{2}\Big(\phi(t_n) + \phi(t_{n+1})\Big)\Big) \Big/ \Big(1 - \frac{\lambda h}{2}\Big).$$

Dans le cas d'un système linéaire d'équations différentielles, nous avons à résoudre un système algébrique linéaire (dans ce cas la méthode est connue sous le nom "Crank–Nicholson"), tandis que dans le cas d'un système non linéaire d'équations différentielles, nous avons un système algébrique non linéaire (voir §2.4 ci-après).

Grâce aux propriétés géométriques de la parabole (la pente d'une sécante est la moyenne des pentes de la parabole aux deux bouts), la règle du trapèze peut être interprétée comme une interpolation par une parabole satisfaisant les pentes du problème aux deux bouts (voir les méthodes de collocation en §2.1.1 ci-après).

Nous voyons en Fig. 4 (images supérieures), que la méthode du trapèze est déjà beaucoup mieux qu'une méthode explicite (au moins les solutions n'explosent pas), mais il reste encore du progrès à faire sur la stabilité, et surtout sur la précision des paraboles, quand le problème devient très raide.

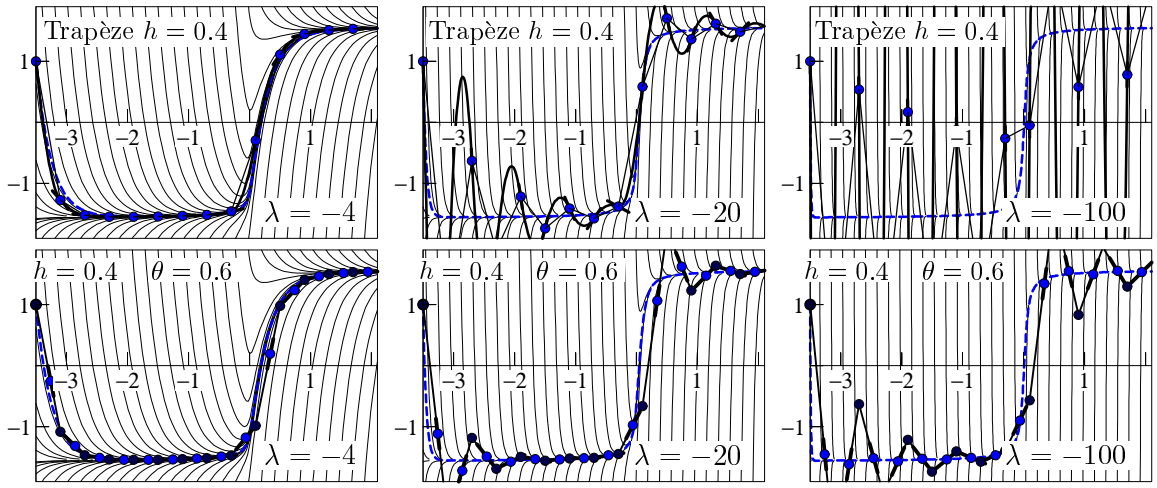

FIGURE 4: Équation de Curtiss–Hirschfelder $\dot{y} = \lambda(y - \arctan 20t)$; valeur initiale $y(-3.5) = 1$. Images du haut : méthode du trapèze à h fixe ; images du bas : méthode thêta.

La méthode thêta. Pour un paramètre $\frac{1}{2} \leq \theta \leq 1$ fixé, cette méthode s'écrit

$$Y_1 = y_n + \theta h f(Y_1)\,, \qquad y_{n+1} = y_n + hf(Y_1) = y_n + \frac{1}{\theta}(Y_1 - y_n)\ .$$

Géométriquement, elle correspond à une interpolation par une droite satisfaisant la pente de l'équation à un point de collocation $t_n + \theta h$. Pour les problèmes très raides, elle se comporte beaucoup mieux (voir Fig. 4). Si θ s'approche de $\frac{1}{2}$, la méthode thêta s'approche de la **règle du point-milieu**, qui possède le même ordre 2 que la règle du trapèze — et les mêmes défauts. Pour $\theta \to 1$, elle s'approche de la **méthode d'Euler implicite** (ou Euler rétrograde)

$$y_{n+1} = y_n + hf(y_{n+1})\ ,$$

de moins en moins précise, mais de plus en plus stable. On choisira un bon compromis $\frac{1}{2} \leq \theta \leq 1$ entre les deux extrêmes, ou mieux encore, une des méthodes Radau IIA (voir §2.1.1 ci-après), qui réunissent les deux avantages.

1.5 Définition et discussion de la A-stabilité

Les phénomènes de stabilité s'expliquent facilement à l'aide de l'équation de Curtiss–Hirschfelder du §1.2. Pour ce problème, nous supposons que $\phi(t)$ ne bouge pas beaucoup et remplaçons $y - \phi(t)$ par y. Cela donne l'équation linéaire

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

Pour un problème général, on linéarise l'équation dans le voisinage d'une solution supposée stationnaire et diagonalise la matrice jacobienne $f'(y)$. Pour chaque composante on obtient ainsi cette même équation, dite **équation test de Dahlquist**, où λ représente une des valeurs propres de la matrice $f'(y)$.

Appliquée à cette équation, la méthode d'Euler explicite devient

$$y_{n+1} = (1 + h\lambda)y_n \qquad \text{ou} \qquad y_{n+1} = R(z)y_n \quad \text{avec} \quad R(z) = 1 + z\,, \quad z = h\lambda\,.$$

On appelle $R(z)$ la **fonction de stabilité** de cette méthode. Pour avoir stabilité (c.-à-d. que la solution numérique n'explose pas), il est nécessaire que

$$|R(z)| \leq 1\,.$$

Cette condition définit le **domaine de stabilité**. Pour la méthode d'Euler explicite, elle devient $|1 + z| = |z - (-1)| \leq 1$ et représente dans le plan complexe un cercle de centre -1 et de rayon 1. Pour avoir stabilité, il faut donc, pour un λ réel, que $-2 \leq h\lambda \leq 0$, ce qui explique les observations de la Fig. 1.

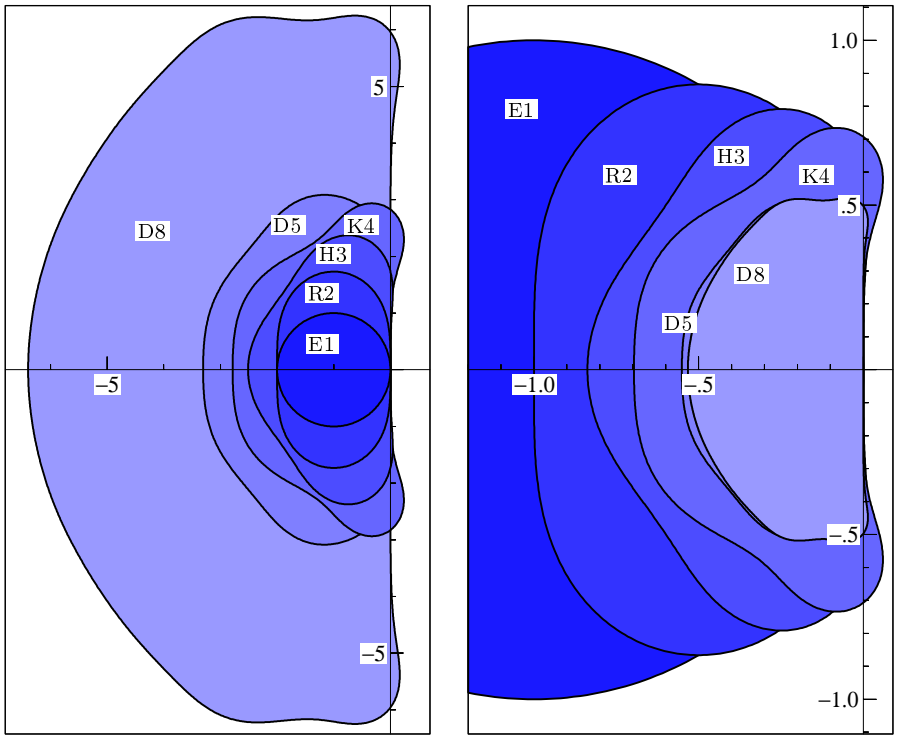

FIGURE 5: Domaines de stabilité des méthodes de Runge-Kutta (à gauche), domaines équitablement réduits (à droite) ; E1=Euler, R2=Runge ordre 2, H3=Heun ordre 3, K4=Kutta ordre 4, D5=Dopri5, D8=Dop853.

Stabilité des méthodes Runge–Kutta. Si l'on applique les méthodes de Runge, Kutta, et DOPRI5 à l'équation linéaire test, on obtient les fonctions de stabilité correspondantes

$$\begin{aligned}
\text{Runge :} \quad & R(z) = 1 + z + \frac{z^2}{2} \\
\text{Kutta :} \quad & R(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} \\
\text{Dopri5 :} \quad & R(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{z^5}{120} + \frac{z^6}{600}\,.
\end{aligned}$$

Ce n'est pas un hasard si beaucoup de ces termes correspondent à ceux de la série exponentielle, car les méthodes approchent la solution exacte, qui est justement $y(h) = e^z y_0$ avec $z = h\lambda$. Les domaines de stabilité correspondants (avec également la méthode de Heun d'ordre 3, et DOP853) sont présentés à gauche dans la Fig. 5. On voit que les méthodes d'Euler, Runge2 et Dopri5 ne sont pas recommandables pour des valeurs propres imaginaires pures (problèmes oscillatoires et hyperboliques), et que les méthodes Heun3, Kutta4 et DOP853 sont applicables aussi bien pour des valeurs propres réelles négatives que pour des valeurs propres imaginaires pures.

Domaines de stabilité réduits. Le domaine de stabilité pour DOP853 dans la Fig. 5 possède une taille impressionante, mais au prix de $s = 12$ évaluations du champs de vecteur $f(y)$ par pas de la méthode. Si nous voulons une comparaison plus équitable des méthodes, il convient de réduire ce domaine par le facteur $1/s$. Les domaines ainsi réduits sont représentés à droite de la Fig. 5 et la figure change considérablement. Un théorème intéressant de Jeltsch–Nevanlinna [JN81] exprime que deux de ces domaines réduits (certaines méthodes multipas inclus) ne sont jamais contenus l'un dans l'autre, autrement dit il n'y a pas de méthode explicite supérieure à une autre méthode explicite.

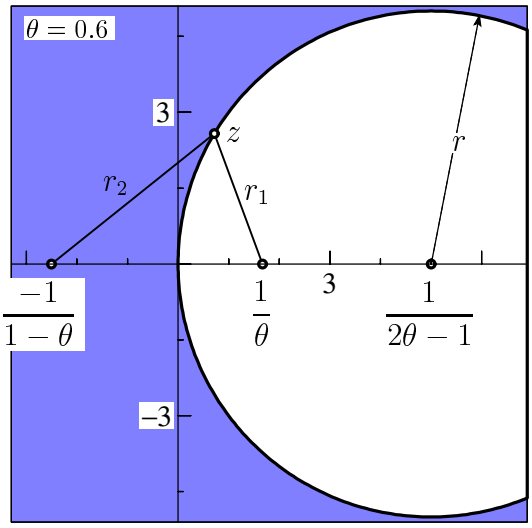

FIGURE 6: Domaine de stabilité de la méthode thêta pour $\theta = 0.6$.

A-stabilité de la méthode thêta. Cette méthode étant implicite, appliquée à l'équation test, elle conduit à une fonction de stabilité rationnelle

$$R(z) = \frac{1 + (1-\theta)z}{1 - \theta z} \quad \text{c.à.d. pour } \theta = \tfrac{1}{2} \text{ et } 1: \quad R(z) = \frac{1 + \frac{1}{2}z}{1 - \frac{1}{2}z} \quad \text{et} \quad R(z) = \frac{1}{1-z}\,.$$

La condition de stabilité devient ici $|1 + (1-\theta)z| \le |1 - \theta z|$, que nous écrivons sous la forme

$$(1-\theta)\,\left|z - \tfrac{-1}{1-\theta}\right| \le \theta\,\left|z - \tfrac{1}{\theta}\right| \qquad \text{ou} \qquad (1-\theta)\,r_2 \le \theta\, r_1$$

(voir Fig. 6). Cela représente l'extérieur d'un cercle de rayon $\frac{1}{2\theta-1}$ tangent à l'axe imaginaire. Nous constatons que tout le demi-plan gauche $\Re z \leq 0$ est contenu dans le domaine de stabilité. Ce demi-plan est l'ensemble des valeurs propres pour lesquelles la solution exacte est stable.

Une méthode est dite **A-stable** si sa fonction de stabilité satisfait $|R(z)| \leq 1$ pour tout z dans le demi-plan gauche $\Re z \leq 0$. La méthode thêta est donc A-stable pour $\frac{1}{2} \leq \theta \leq 1$. Pour $\theta < \frac{1}{2}$, le cercle passe dans le demi-plan gauche et la méthode cesse d'être A-stable.

Méthodes du point milieu et du trapèze. Si $\theta \to \frac{1}{2}$, la méthode thêta tend vers la méthode du point milieu (ayant la même fonction de stabilité que la règle du trapèze) et le domaine de stabilité tend vers le demi-plan gauche. La méthode est donc théoriquement A-stable, mais pour des problème très raides, où $z \to -\infty$, nous avons $R(z) \to -1$. Cela explique le très mauvais amortissement des erreurs dans les images du haut de la Fig. 4.

L-stabilité de la méthode d'Euler implicite. Par contre, si $\theta = 1$, nous avons, en plus de la A-stabilité, la propriété $R(-\infty) = 0$, et donc un amortissement maximal des erreurs pour les problèmes très raides, et la méthode est dite **L-stable**. L'importance pratique de cette propriété a été découverte par Ehle [Ehl69] et Prothero–Robinson [PR74].

2 Intégrateurs numériques d'ordre supérieur

Les méthodes numériques comme la règle du trapèze, la règle du point-milieu et surtout la méthode d'Euler implicite (voir le §1.4) sont certes adaptées à la résolution des problèmes raides, mais leur grand défaut est leur manque de précision. En pratique, on a besoin de méthodes d'ordre supérieur qui toutefois restent efficaces pour des problèmes raides. Nous allons discuter des méthodes de Runge–Kutta implicites, des BDF (méthodes multipas) ainsi que des méthodes d'extrapolation. Plus de détails et d'autres classes d'intégrateurs peuvent être trouvés dans le livre [HW96].

2.1 Méthodes de Runge–Kutta implicites

Parmi les méthodes à un pas, les méthodes de Runge–Kutta implicites sont les plus importantes. Nous considérons deux cas particuliers.

2.1.1 Méthodes de collocation

Considérons une équation différentielle $\dot{y} = f(y)$ et supposons connaître une approximation y_n de la solution $y(t_n)$. Pour des nombres $0 \leq c_1 < c_2 < \cdots < c_s \leq 1$ donnés, nous cherchons un polynôme $u(t)$ de degré s qui vérifie

$$u(t_n) = y_n \qquad \text{et} \qquad \dot{u}(t_n + c_i h) = f(u(t_n + c_i h)), \quad i = 1, \ldots, s.$$

L'approximation de $y(t_{n+1}) = y(t_n + h)$ est alors donnée par $y_{n+1} = u(t_n + h)$.

L'idée de la collocation (c.-à-d., satisfaire l'équation à un nombre fini de points, les points de collocation) se retrouve souvent dans la littérature numérique. Pour les équations différentielles ordinaires elle est due à [GS69] et [Wri70].

Equivalence avec des méthodes de Runge–Kutta implicites. Posons $Y_i = u(t_n + c_i h)$ et exprimons la dérivée $\dot{u}(t)$ en termes de $f(Y_i)$. Après une intégration de ce polynôme, les conditions de collocation deviennent équivalentes à

$$\begin{aligned} Y_i &= y_n + h \sum_{j=1}^{s} a_{ij} f(Y_j), \qquad i = 1, \ldots, s \\ y_{n+1} &= y_n + h \sum_{i=1}^{s} b_i f(Y_i) \end{aligned}$$

où les coefficients a_{ij} et b_i sont déterminés par

$$\begin{aligned} \sum_{j=1}^{s} a_{ij} c_j^{k-1} &= \frac{c_i^k}{k}, \qquad k = 1, \ldots, s, \quad i = 1, \ldots, s \\ \sum_{i=1}^{s} b_i c_i^{k-1} &= \frac{1}{k}, \qquad k = 1, \ldots, s. \end{aligned}$$

Cette reformulation ne montre pas seulement l'existence de la solution numérique pour h suffisament petit, mais elle est aussi utile pour la programmation de ces méthodes. La grande difficulté de la programmation se trouve dans la résolution du système non linéaire pour les inconnues $Y_i, i = 1, \ldots, s$.

Les intégrateurs vus au §1.4 sont tous des méthodes de collocation. La régle du trapèze est obtenue avec $s = 2$ et $c_1 = 0$, $c_2 = 1$, la règle du point-milieu avec $s = 1$ et $c_1 = 1/2$, et la méthode d'Euler implicite avec $s = 1$ et $c_1 = 1$.

Ordre et superconvergence. Pour le problème $\dot{y} = f(t)$, la méthode de Runge–Kutta devient une formule de quadrature $\int_{t_n}^{t_n+h} f(t)\,dt \approx h \sum_{i=1}^{s} b_i\, f(t_n + c_i h)$. Cette formule de quadrature possède l'ordre p (c.-à-d., elle intègre les polynômes de degré $\leq p-1$ de manière exacte) si ses coefficients vérifient

$$\sum_{i=1}^{s} b_i c_i^{k-1} = \frac{1}{k}, \qquad k = 1, \ldots, p.$$

Un résultat intéressant [GS69] est que sous cette condition, une méthode de collocation possède également l'ordre p pour toute équation différentielle et pas seulement pour les problèmes de quadrature.

En choisissant les c_i comme les zéros du polynôme de Legendre $\frac{d^s}{dx^s}(x^s(1-x)^s)$, on obtient une méthode numérique d'ordre $p = 2s$ (méthode de Gauss). Parmi les méthodes de Runge–Kutta à s étages, c'est l'unique méthode avec l'ordre maximal possible.

Fonction de stabilité. En appliquant une méthode de Runge–Kutta implicite au problème de test linéaire $\dot{y} = \lambda y$, l'approximation numérique est donnée par

$$y_{n+1} = R(h\lambda) y_n \qquad R(z) = 1 + z b^T (I - zA)^{-1} \mathbb{1}$$

où $b = (b_i)_{i=1}^s$, $A = (a_{ij})_{i,j=1}^s$ et $\mathbb{1} = (1, \ldots, 1)^T$. La fonction $R(z)$ s'appelle la fonction de stabilité. Pour avoir A-stabilité (voir le §1.5), il faut satisfaire $|R(z)| \leq 1$ pour tout $z \in \mathbb{C}$ dans le demi-plan gauche $\Re z \leq 0$.

Pour beaucoup de méthodes de Runge–Kutta avec un ordre élevé, la fonction de stabilité est une approximation de Padé de la fonction exponentielle, c.-à-d.,

$$R_{kj}(z) = \frac{P_{kj}(z)}{Q_{kj}(z)},$$

où $Q_{kj}(z) = P_{jk}(-z)$ et $P_{kj}(z)$ est le polynôme de degré k donné par

$$P_{kj}(z) = 1 + \frac{k}{j+k} \cdot z + \frac{k(k-1)}{(j+k)(j+k-1)} \cdot \frac{z^2}{2!} + \ldots + \frac{k(k-1)\cdots 1}{(j+k)\cdots(j+1)} \cdot \frac{z^k}{k!}.$$

La fraction rationelle $R_{kj}(z)$ est A-stable si et seulement si $k \leq j \leq k+2$ [WHN78].

L'unique méthode d'ordre $p = 2s$ (méthode de Gauss) possède l'approximation de Padé diagonale $R_{ss}(z)$ comme fonction de stabilité. Elle est donc A-stable, mais son

défaut est que $R_{ss}(\infty) = \pm 1$ et il n'a pas suffisamment d'amortissement dans l'erreur correspondant au modes le plus raides.

Précision pour des problèmes très raides ("stiff accuracy"). Pour des problèmes à perturbation singulière $\dot{y} = f(y,z)$, $\varepsilon\dot{z} = g(y,z)$ avec $0 < \varepsilon \ll 1$ très petit et sous une condition de stabilité pour l'équation différentielle, la solution exacte s'approche rapidement d'une sous-variété qui est $\mathcal{O}(\varepsilon)$-proche de $g(y,z) = 0$. Si la matrice $A = (a_{ij})$ de la méthode de Runge–Kutta est inversible, les étages internes Y_i possèdent la même propriété. En imposant la condition

$$b_i = a_{si} \qquad \text{pour} \quad i = 1, \ldots, s,$$

la solution numérique devient un étage interne et hérite donc de cette propriété d'approximation. Si $0 < c_1 < \cdots < c_s = 1$, la méthode de collocation satisfait cette condition, et est donc bien adaptée pour approcher numériquement la solution de problèmes à perturbation singulière.

Les méthodes Radau IIA. Parmi les méthodes de Runge–Kutta implicites, celles de type Radau IIA ont des propriétés très favorables pour résoudre des problèmes raides. Il s'agit des méthodes de collocation où les c_i sont les zéros du polynôme

$$\frac{d^{s-1}}{dx^{s-1}}\Big(x^{s-1}(1-x)^s\Big).$$

Ces zéros sont distincts et tous dans l'intervalle $(0,1]$. Ils sont strictement positifs et la matrice $A = (a_{ij})$ est donc inversible. Comme $c_s = 1$, les coefficients satisfont $b_i = a_{si}$ pout tout i. Pour les cas $s = 2$ et $s = 3$, les matrices des coefficients sont données dans le tableau 1, les $c_i = \sum_j a_{ij}$ (somme des lignes de la matrice) sont à gauche et les poids $b_i = a_{si}$ en-dessous.

La formule de quadrature construite à partir des abscisses $c_1, \ldots, c_{s-1}, c_s = 1$ s'appelle une formule de Radau. Elle est d'ordre $p = 2s - 1$ et donc d'ordre maximal parmi les formules de quadrature satisfaisant $c_s = 1$.

La fonction de stabilité de méthodes Radau IIA est l'approximation de Padé $R_{s-1,s}(z)$. Elle est A-stable. Comme $R_{s-1,s}(\infty) = 0$, elle est aussi L-stable.

Tableau 1: Méthodes Radau IIA d'ordre 3 et 5

$$\begin{array}{c|cc}
\frac{1}{3} & \frac{5}{12} & -\frac{1}{12} \\
1 & \frac{3}{4} & \frac{1}{4} \\
\hline
 & \frac{3}{4} & \frac{1}{4}
\end{array}
\qquad
\begin{array}{c|ccc}
\frac{4-\sqrt{6}}{10} & \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\
\frac{4+\sqrt{6}}{10} & \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\
1 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \\
\hline
 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9}
\end{array}$$

2.1.2 Méthodes de Runge–Kutta diagonalement implicites

Si on veut programmer une méthode de Runge–Kutta implicite dont tous les coefficients a_{ij} sont non-nuls (par exemple Radau IIA), on est confronté à la difficulté de devoir résoudre un grand système non linéaire d'inconnues $Y_1, \ldots, Y_s$. En renonçant à l'ordre maximal (super-convergence) on peut considérer des méthodes

$$\begin{aligned} Y_i &= y_n + h \sum_{j=1}^{i} a_{ij} f(Y_j), \qquad i = 1, \ldots, s \\ y_{n+1} &= y_n + h \sum_{i=1}^{s} b_i f(Y_i) \end{aligned}$$

où la somme dans la définition de Y_i s'arrête avec l'indice i. Par conséquent, les Y_i peuvent être calculés l'un après l'autre en résolvant chaque fois un système non linéaire $Y_i = \eta_i + h a_{ii} f(Y_i)$ de même dimension que y. La résolution de cette suite de systèmes non linéaires est rendue encore plus efficace si tous les $a_{ii} = \gamma$ prennent la même valeur.

Étude de l'ordre. Après cette simplification du système non linéaire, on n'a plus une étude de l'ordre aussi élégante que précédemment. On est obligé de développer la solution exacte de l'équation différentielle et la solution numérique en séries de Taylor (autour de $h = 0$, comme pour des méthodes de Runge–Kutta explicites) et de comparer les puissances de h jusqu'au terme h^p. Ceci donne pour ordre $p = 3$ les conditions

$$\sum_{i=1}^{s} b_i = 1, \qquad \sum_{i=1}^{s} b_i c_i = \frac{1}{2}, \qquad \sum_{i=1}^{s} b_i c_i^2 = \frac{1}{3}, \qquad \sum_{i=1}^{s} b_i \sum_{j=1}^{i} a_{ij} c_j = \frac{1}{6}.$$

Le nombre de conditions pour les coefficients b_i, a_{ij}, c_j augmente exponentiellement avec l'ordre p et il est donc difficile de construire des méthodes diagonalement implicites avec un ordre $p > 5$.

Fonction de stabilité. En supposant $a_{ii} = \gamma$ pour tout i, la fonction de stabilité prend la forme

$$R(z) = \frac{P(z)}{(1 - \gamma z)^s}.$$

TABLEAU 2: A-stabilité de méthodes diagonalement implicites d'ordre $p \geq s$

s	A-stabilité	A-stabilité et $p = s + 1$	L-stabilité
1	$1/2 \leq \gamma < \infty$	$1/2$	1
2	$1/4 \leq \gamma < \infty$	$(3 + \sqrt{3})/6$	$(2 \pm \sqrt{2})/2$
3	$1/3 \leq \gamma \leq 1.06857902$	1.06857902	0.43586652
4	$0.39433757 \leq \gamma \leq 1.28057976$	–	0.57281606

Si l'ordre de la méthode est $p \geq s$, on a $R(z) = e^z + \mathcal{O}(z^{s+1})$ et donc le polynôme $P(z) = e^z(1-\gamma z)^s + \mathcal{O}(z^{s+1})$ de degré s est uniquement déterminé par le coefficient γ. Le tableau 2 indique les valeurs de γ pour lesquelles on obtient une méthode A-stable, une méthode A-stable avec ordre $p = s+1$, ou bien une méthode L-stable. Si on impose la condition $b_i = a_{si}$ pour tout i, la méthode sera automatiquement L-stable. Cette condition est importante pour la résolution de problèmes très raides.

Example. Pour une méthode avec $s = 3$ étages, qui satisfait la condition $b_i = a_{si}$ pour tout i, on a quatre coefficients (γ, c_2, b_1, b_2) à déterminer pour satisfaire les quatre conditions d'ordre $p = 3$. Il existe une seule méthode L-stable qui satisfait ces conditions. Elle est donnée dans le tableau 3. La valeur de γ est obtenue comme zéro du polynôme $6x^3 - 18x^2 + 9x - 1 = 0$.

TABLEAU 3: Une méthode L-stable d'ordre $p = 3$

$$\begin{array}{c|ccc} \gamma & \gamma & 0 & 0 \\ \frac{1+\gamma}{2} & \frac{1-\gamma}{2} & \gamma & 0 \\ 1 & b_1(\gamma) & b_2(\gamma) & \gamma \\ \hline & b_1(\gamma) & b_2(\gamma) & \gamma \end{array} \qquad \begin{aligned} \gamma &= 0.4358665215084590 \\ b_1(\gamma) &= \tfrac{1}{4}\big(-1 + 16\gamma - 6\gamma^2\big) \\ b_2(\gamma) &= \tfrac{1}{4}\big(5 - 20\gamma + 6\gamma^2\big) \end{aligned}$$

2.2 Méthodes multipas (BDF)

Les intégrateurs numériques pour $\dot{y} = f(y)$ qui exploitent l'information de plusieurs pas consécutifs sont les méthodes à pas multiples (méthodes multipas). Pour une application à pas constant h, elles sont définies par

$$\sum_{i=0}^{k} \alpha_i \, y_{n+i} = h \sum_{i=0}^{k} \beta_i \, f(y_{n+i})$$

où $|\alpha_0| + |\beta_0| > 0$ et $\alpha_k \neq 0$, de sorte que l'approximation y_{n+k} de $y(t_{n+k})$ peut être calculée à partir de $y_n, \ldots, y_{n+k-1}$. Cette méthode est caractérisée par les polynômes générateurs

$$\rho(\zeta) = \sum_{i=0}^{k} \alpha_i \, \zeta^i, \qquad \sigma(\zeta) = \sum_{i=0}^{k} \beta_i \, \zeta^i.$$

Pour des problèmes raides, on considère des méthodes implicites (c.-à-d., $\beta_k \neq 0$), qui nécessitent la résolution d'un système non linéaire $y_{n+k} = \eta_{n+k} + h\gamma_k f(y_{n+k})$ pour y_{n+k}, où $\gamma_k = \beta_k/\alpha_k$ et η_{n+k} est une expression connue.

Ordre et zéro-stabilité. On dit qu'une méthode multipas possède l'ordre p, si pour chaque fonction lisse $y(t)$ on a $\sum_{i=0}^{k} \alpha_i \, y(t_{n+i}) - h \sum_{i=0}^{k} \beta_i \, \dot{y}(t_{n+i}) = \mathcal{O}(h^{p+1})$. Cette

propriété ne suffit pas encore pour que la solution numérique converge vers la solution exacte du problème. Il faut en plus que la méthode soit zéro-stable, ce qui signifie que tous les zéros du polynôme $\rho(\zeta)$ sont bornés par 1, et ceux sur le cercle unité sont simples.

A-stabilité et domaine de stabilité. Pour le problème de test $\dot{y} = \lambda y$, la méthode multipas devient l'équation aux différences $\sum_{i=0}^{k}(\alpha_i - \mu\beta_i)\, y_{n+i} = 0$ avec $\mu = \lambda h$. Sa solution y_n est une combinaison linéaire des $\zeta_i(\mu)^n$ (multipliée par un polynôme en n en cas de racines multiples), où $\zeta_1(\mu), \ldots, \zeta_k(\mu)$ sont les racines du polynôme caractéristique

$$\rho(\zeta) - \mu\, \sigma(\zeta) = 0.$$

La solution y_n reste bornée pour $n \to \infty$ si μ est dans le domaine de stabilité

$$S = \left\{ \mu \in \mathbb{C}\ ; \quad \begin{array}{l} \text{toutes les racines } \zeta_i(\mu) \text{ satisfont } |\zeta_i(\mu)| \le 1, \\ \text{et les racines multiples satisfont } |\zeta_i(\mu)| < 1 \end{array} \right\}$$

Il est souhaitable que le domaine de stabilité contienne le demi-plan gauche. Dans ce cas la méthode est dite A-stable. Un résultat très célèbre de G. Dahlquist [Dah63] est que l'ordre maximal d'une méthode multipas A-stable est 2.

La condition de A-stabilité est très restrictive, et dans beaucoup situations pratiques il suffit qu'un secteur $S_\alpha = \{\mu \in \mathbb{C}\,;\ |\arg(-\mu)| < \alpha\}$ avec α proche de $\pi/2$ soit contenu dans le domaine de stabilité. Dans ce cas, la méthode est dite A(α)-stable.

Les méthodes BDF. Si $\mu \to \infty$, les zéros de $\rho(\zeta) - \mu\, \sigma(\zeta)$ tendent vers les zéros de $\sigma(\zeta)$. Pour avoir un bon amortissement des composantes très raides, il faut que $\zeta_i(\infty) = 0$ pour tout i. Ceci est possible seulement si $\sigma(\zeta) = \beta_k \zeta^k$ avec $\beta_k \neq 0$ (c.-à-d., $\beta_i = 0$ pour $i < k$). Une fois les coefficients β_i fixés, on peut déterminer les coefficients α_i pour obtenir l'ordre $p = k$. Les formules ainsi obtenues peuvent être écrites à l'aide des différences rétrogrades (d'où le nom "backward differentiation formulae") sous la forme

$$\sum_{j=1}^{k} \frac{1}{j}\, \nabla^j y_{n+k} \;=\; h\, f(y_{n+k}).$$

Rappelons que $\nabla y_{n+k} = y_{n+k} - y_{n+k-1}$, et récursivement $\nabla^{j+1} y_{n+k} = \nabla^j y_{n+k} - \nabla^j y_{n+k-1}$.

Pour $k = 1$ on retrouve la méthode d'Euler implicite, et pout $k = 2$ on obtient

$$\frac{3}{2}\, y_{n+2} - 2\, y_{n+1} + \frac{1}{2}\, y_n \;=\; h\, f(y_{n+2}).$$

Tableau 4: A(α)-stabilité des méthodes BDF

k	1	2	3	4	5	6
α	90°	90°	86.03°	73.35°	51.84°	17.84°

Ces deux méthodes sont A-stables. Pour $3 \leq k \leq 6$, les méthodes obtenues sont A(α)-stables avec des angles α donnés dans le tableau 4. Pour $k > 6$ un des zéros du polynôme $\rho(\zeta)$ possède un module strictement plus grand que 1. Les méthodes correspondantes ne sont pas zéro-stables et donc pas utiles pour l'intégration numérique.

Pour dessiner le domaine de stabilité d'une méthode multipas, l'observation suivante est très utile : le point μ est au bord ∂S du domaine de stabilité si une racine de $\rho(\zeta) - \mu\,\sigma(\zeta) = 0$ possède un module 1. Ceci implique que le bord ∂S est contenu dans la courbe donnée par ("root locus curve")

$$\gamma(t) = \frac{\rho(e^{it})}{\sigma(e^{it})}, \qquad 0 \leq t \leq 2\pi.$$

Ces courbes sont dessinées pour les méthodes BDF d'ordre $k = 1, \ldots, 6$ dans la Fig. 7. Pour $k = 1$ c'est un cercle de rayon 1 centré au point 1, et le point d'intersection avec l'axe réel croit avec k. Le domaine de stabilité est l'extérieur de la courbe.

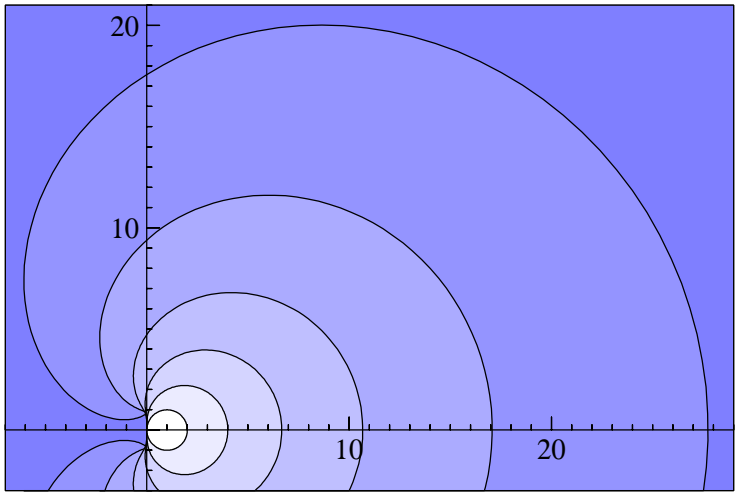

FIGURE 7: Domain de stabilité pour les méthodes BDF d'ordre $k = 1, \ldots, 6$.

2.3 Méthodes d'extrapolation

L'extrapolation est une technique qui permet de créer, à partir d'une méthode de base très simple, des approximations de grande précision. Pour des équations différentielles raides, un bon choix de la méthode de base [Deu85] est la méthode d'Euler linéairement implicite

$$y_{i+1} = y_i + h\,(I - hJ)^{-1} f(y_i)$$

où $J \approx f'(y_0)$ est une approximation de la matrice jacobienne de $f(y)$. Pour un problème linéaire $\dot{y} = Jy$, elle est équivalente à la méthode d'Euler implicite. L'idée d'extrapolation repose sur le fait que l'erreur globale de cet intégrateur possède un développement asymptotique en puissances de h.

Prenons une suite de nombres entiers $n_1 < n_2 < n_3 < \ldots$; par exemple $n_j = j$. Notons ensuite T_{j1} l'approximation de $y(t_0 + H)$ obtenue en appliquant la méthode de base n_j fois avec $h = h_j = H/n_j$. Ensuite, avec l'algorithme de Aitken–Neville

$$T_{j,k+1} = T_{j,k} + \frac{T_{j,k} - T_{j-1,k}}{n_j/n_{j-k} - 1}$$

on peut éliminer les premiers termes du développement asymptotique et on obtient ainsi un tableau des approximations de $y(t_0 + H)$:

$$\begin{array}{cccc} T_{11} & & & \\ T_{21} & T_{22} & & \\ T_{31} & T_{32} & T_{33} & \\ \vdots & \vdots & \vdots & \ddots \end{array}$$

Les approximations dans la première colonne sont d'ordre 1, celles de la deuxième colonne d'ordre 2, et celles de la $k$ième colonne d'ordre k.

A(α)-stabilité. En appliquant cette procédure à l'équation de test $\dot{y} = \lambda y$ on obtient $T_{jk} = R_{jk}(h\lambda)y_0$ où la fonction de stabilité de l'approximation T_{jk} est obtenue par

$$R_{j1}(z) = \left(1 - \frac{z}{n_j}\right)^{-n_j}, \qquad R_{j,k+1}(z) = R_{j,k}(z) + \frac{R_{j,k}(z) - R_{j-1,k}(z)}{n_j/n_{j-k} - 1}.$$

On peut vérifier que les approximations $R_{j1}(z)$ et $R_{j2}(z)$ sont toutes A-stables, les autres approximations sont $A(\alpha)$-stables avec α très proche de $\pi/2$ (voir le tableau 5).

Tableau 5: A(α)-stabilité des méthodes d'extrapolation

90°					
90°	90°				
90°	90°	89.85°			
90°	90°	89.90°	89.77°		
90°	90°	89.93°	89.84°	89.77°	
90°	90°	89.95°	89.88°	89.84°	89.77°

2.4 Programmation des intégrateurs numériques

La programmation des intégrateurs numériques pour des équations différentielles raides est beaucoup moins évidente que pour des problèmes non-raides où on utilise en général des méthodes explicites. Il existe quelques méthodes pour des problèmes raides qui évitent des équations non linéaires (par exemple la méthode d'extrapolation discutée

au §2.3 précédent) mais la plupart des méthodes utilisées en pratique sont implicites et nécessitent donc la résolution d'un système non linéaire.

Résolution du système non linéaire. Pour les méthodes implicites du §1.4, pour les méthodes de Runge–Kutta diagonalement implicites, et pour les méthodes multipas (BDF), on est amené à la résolution d'un système de la forme

$$y = \eta + h\gamma f(y)$$

où η (un vecteur) et $h\gamma > 0$ sont donnés. On ne peut pas appliquer une itération de point-fixe, car ceci restreindrait la longueur du pas h de manière inacceptable (notons que les valeur propres de la matrice jacobienne $f'(y)$ sont très grandes et que l'itération converge seulement si $h\gamma < \|f'(y)\|^{-1}$). On est obligé d'appliquer la méthode de Newton ou une variante. Pour réduire l'influence des erreurs d'arrondi, il est recommandé de travailler avec le vecteur $z = y - \eta$ (qui, en général, est plus petit en norme que le vecteur y) et d'appliquer la méthode de Newton simplifiée au système $F(z) = z - h\gamma f(\eta + z) = 0$. Ceci donne l'itération

$$z^{k+1} = z^k + \Delta z^k, \qquad (I - h\gamma J)\Delta z^k = -z^k + h\gamma f(\eta + z^k)$$

où $J \approx f'(\eta)$. Pour l'initialisation z^0 on peut prendre le vecteur η ou une autre approximation de la solution cherchée. L'utilisation de la même matrice J pour toutes les itérations rend cet algorithme efficace.

Pour une méthode de Runge–Kutta implicite avec plusieurs étages (par exemple Radau IIA) on est conduit à un système non linéaire de dimension $n \cdot s$ où n est la dimension de y et s le nombre d'étages. Pour une programmation efficace il faut exploiter la structure de la matrice jacobienne (voir [HW96, §IV.8]).

3 Méthodes pour des problèmes particuliers

En pratique, on rencontre souvent des équations différentielles raides de très grande taille pour lesquelles la résolution numérique du système non linéaire de la méthode implicite est très couteuse ou même impossible. Il arrive aussi que la raideur de l'équation différentielle soit présente seulement dans une petite partie de l'équation, on aimerait alors tirer profit de cette situation. Ce paragraphe présente quelques approches intéressantes pour ces problèmes particuliers.

3.1 Méthodes explicites avec longues régions de stabilité

Selon le théorème de Jeltsch–Nevanlinna évoqué au §1.5, il n'existe pas de méthode explicite qui soit supérieure, du point de vue de la stabilité, à une autre méthode explicite pour tous les problèmes. Par contre, en présence de l'information sur la localisation des valeurs propres, on peut désigner des méthodes efficaces pour cette classe de problèmes. La classe la plus importante est celle des problèmes où les valeurs propres sont réelles et localisées sur l'axe réel négatif. Comme référence on pourra consulter [HV03, chap. V] et [HW96, §IV.2].

3.1.1 Méthodes de Runge–Kutta–Chebyshev

Commençons par chercher la meilleure fonction de stabilité. Pour une méthode d'ordre (au moins) 1, explicite à s étages, $R(z)$ est un polynôme de degré s de la forme $R(z) = 1 + z + \mathcal{O}(z^2)$. Le polynôme de Chebyshev $T_s(t)$ étant équioscillant entre -1 et $+1$ sur l'intervalle $-1 \leq t \leq 1$, avec $T'(1) = s^2$, il fournit le polynôme désiré

$$R(z) = T_s(1 + z/s^2)$$

avec un intervalle de stabilité $-2s^2 \leq z \leq 0$ le plus long possible (voir la première image de la Fig. 8).

Réalisation comme méthode de Runge–Kutta. Il faut maintenant trouver un algorithme pour équations différentielles qui possède cette fonction de stabilité. Il y a deux possibilités :

- par **factorisation** de $R(z) = \prod_{i=1}^{s}(1 + \delta_i z)$ et une composition de pas d'Euler

  $$Y_0 := y_n, \qquad Y_i := Y_{i-1} + h\delta_i f(Y_{i-1}), \quad (i = 1, 2, \ldots, s), \qquad y_{n+1} := Y_s$$

  (méthode proposée par Guillou–Lago [GL61], élaborée par Lebedev [Leb89]). Elle a le désavantage que certains de ses pas sont très grands et que la méthode est très sensible à l'ordre de grandeur des δ_i ;

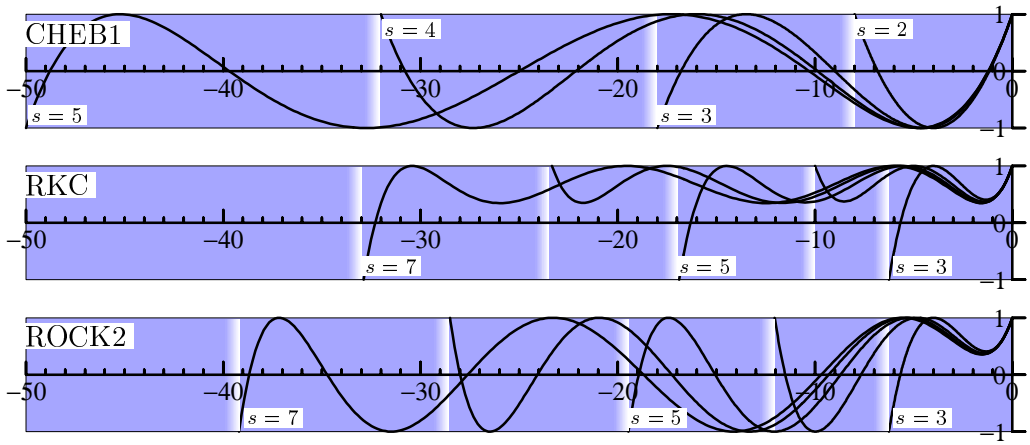

FIGURE 8: Fonctions de stabilité des méthodes de Chebyshev

- par **récurrence** en utilisant $T_j(t) = 2tT_{j-1}(t) - T_{j-2}(t)$. Ainsi les grandeurs $R_j(z) = T_j(1 + z/s^2)$ satisfont $R_j(z) = \frac{2z}{s^2}R_{j-1}(z) + 2R_{j-1}(z) - R_{j-2}(z)$. On obtient alors la méthode suivante

$$Y_0 = y_n, \quad Y_1 = Y_0 + \frac{h}{s^2}f(Y_0), \quad Y_j = \frac{2h}{s^2}f(Y_{j-1}) + 2Y_{j-1} - Y_{j-2}, \quad y_{n+1} := Y_s,$$

proposée par van der Houwen–Sommeijer [HS80] et Sommeijer–Verwer [SV80].

Méthode d'ordre 2 (code RKC). Comme $T_s''(1) = s^2(s^2-1)/3$, la fonction $R(z)$ ci-dessus ne satisfait pas la condition d'ordre 2, qui est $R''(0) = 1/2$. On peut corriger cela par une transformation affine

$$R(z) = a_s + b_s T_s(1 + w_s z), \qquad w_s = \frac{3}{s^2-1}, \quad b_s = \frac{s^2-1}{3s^2}, \quad a_s = 1 - b_s.$$

Cette formule améliorée par un facteur d'amortissement, et la formule de récurrence ci-dessus modifiée, constituent la base du code RKC d'ordre 2 [SSV98].

3.1.2 Approche de Lebedev, Medovikov et Abdulle (code ROCK)

Comme le montre la deuxième image de la Fig. 8, les intervalles de stabilité pour RKC ne sont pas optimaux parmi les approximations d'ordre 2. Mais si l'on veut utiliser les polynômes optimaux (troisième image), on perd la belle formule récursive ci-dessus. L'idée (due à Abdulle et Medovikov [AM01]) est alors d'approcher ces polynômes par une suite de polynômes orthogonaux

$$R_s(z) = w_p(z)P_{s-p}(z), \qquad P_j(x) \text{ pol. orthog. par rapport à } \frac{w_p^2(x)}{\sqrt{1-x^2}}.$$

Grâce à un théorème de Bernstein (1930), ces polyômes vont être presque optimaux, et grâce à l'orthogonalité, il existe une formule récursive de calcul. Nous obtenons

alors une méthode, qui possède les deux avantages, un intervalle de stabilité presque optimal, et une relation de récurrence. De plus, nous pouvons déterminer le polynôme $w_p(z)$ de façon à ce que $R_s(z)$ possède l'ordre p.

Réalisation comme méthode de Runge–Kutta. Pour l'ordre 4, une méthode de Runge–Kutta n'a plus automatiquement le même ordre que sa fonction de stabilité, car des conditions supplémentaires pour les problèmes non linéaires sont à satisfaire. Abdulle et Medovikov résolvent ce problème à l'aide d'une théorie très élaborée, la théorie du **groupe de Butcher**. Le code correspondant, ROCK4, fournit d'excellents résultats dans des problèmes très complexes en calcul scientifique [Abd02]. Ce code a produit la solution dessinée en Fig. 3 du §1.3.

3.2 Méthodes de séparation (splitting)

Considérons une équation différentielle de la forme

$$\dot{y} = f^{[1]}(y) + f^{[2]}(y)$$

et supposons que la résolution des systèmes partiels $\dot{y} = f^{[1]}(y)$ et $\dot{y} = f^{[2]}(y)$ soit beaucoup plus simple que celle du système entier. Notons par $\varphi_t(y)$ le flot exact de l'équation différentielle (c.-à-d., l'application $y_0 \mapsto \varphi_t(y_0)$ représentant la solution exacte $y(t)$ au temps t pour la valeur initiale $y(0) = y_0$) et par $\varphi_t^{[1]}(y)$ et $\varphi_t^{[2]}(y)$ les flots des sytèmes partiels. L'idée est d'approcher la solution exacte $\varphi_t(y)$ par une composition $(\varphi_t^{[1]} \circ \varphi_t^{[2]})(y)$ ou $(\varphi_t^{[2]} \circ \varphi_t^{[1]})(y)$ (Lie–Trotter splitting), ou mieux encore par une version symétrique

$$(\varphi_{t/2}^{[1]} \circ \varphi_t^{[2]} \circ \varphi_{t/2}^{[1]})(y) \qquad \text{ou} \qquad (\varphi_{t/2}^{[2]} \circ \varphi_t^{[1]} \circ \varphi_{t/2}^{[2]})(y).$$

Cette approximation est connue sous le nom de **Strang splitting** [Str68], mais elle peut également être appelée **Marchuk splitting** [Mar68] ou **méthode des pas fractionnaires** [Tem68]. Ces compositions représentent la solution exacte $\varphi_t(y)$ uniquement dans la situation où les deux flots $\varphi_t^{[1]}(y)$ et $\varphi_t^{[2]}(y)$ commutent (ce qui est rarement le cas). Dans le cas général on introduit une erreur qui est $(\varphi_h^{[1]} \circ \varphi_h^{[2]})(y) - \varphi_h(y) = \mathcal{O}(h^2)$ pour le Lie-Trotter splitting et $(\varphi_{h/2}^{[2]} \circ \varphi_h^{[1]} \circ \varphi_{h/2}^{[2]})(y) - \varphi_h(y) = \mathcal{O}(h^3)$ pour le Strang splitting (vérification par un développement en série de Taylor).

Si les flots exacts $\varphi_t^{[1]}(y)$ et $\varphi_t^{[2]}(y)$ sont connus explicitement, alors chacune de ces compositions donne une méthode numérique pour la résolution du problème original. Sinon, on peut remplacer les flots exacts $\varphi_t^{[i]}(y)$ par des approximations numériques $\Phi_t^{[i]}(y)$, et ensuite considérer l'intégrateur numérique $\Phi_h(y) = (\Phi_{h/2}^{[2]} \circ \Phi_h^{[1]} \circ \Phi_{h/2}^{[2]})(y)$.

Exemple 3.1 (problème de réaction-diffusion) Une réaction chimique célèbre convergeant vers une solution périodique est la réaction "Oregonator" déjà étudiée au

paragraphe §1.3. En ajoutant de la diffusion nous obtenons après discrétisation en espace le système (pour $i = 0, 1, \ldots, n$)

$$\begin{aligned}
\dot{a}_i &= \delta^2 a_i + 77.27(b_i + a_i\,(1 - 8.375 \cdot 10^{-6} a_i - b_i)) \\
\dot{b}_i &= \delta^2 b_i + (c_i - (1 + a_i)\, b_i)/77.27 \\
\dot{c}_i &= \delta^2 c_i + 0.161(a_i - c_i)
\end{aligned}$$

où $\delta^2 a_i = (a_{i+1} - 2a_i + a_{i-1})/\Delta x^2$ avec $\Delta x = 1/n$ est une approximation du Laplacien, et les relations $a_{-1} = a_1$ et $a_{n+1} = a_{n-1}$ simulent des conditions de Neumann au bord de l'intervalle $[0, 1]$. Comme valeurs initiales nous prenons $a_i(0) = 1 + x_i$, $b_i(0) = \exp(-300(x_i - 0.5)^2)$ et $c_i(0) = 0.5 + 10\, x_i(1 - x_i)$, où $x_i = i\,\Delta x$. En rassemblant dans le vecteur y toutes les composantes de a_i, b_i et c_i, le problème est de la forme $\dot{y} = f^{[1]}(y) + f^{[2]}(y)$, où $f^{[1]}(y)$ représente la diffusion et $f^{[2]}(y)$ la réaction. Le problème $\dot{y} = f^{[2]}(y)$ est constitué de $n + 1$ systèmes séparés de dimension 3, qui peuvent être résolus sans difficultés avec des programmes existants (par exemple avec RADAU5). L'autre problème $\dot{y} = f^{[1]}(y)$ est une équation de la chaleur pour laquelle il existe des algorithmes très efficaces, aussi en dimension 2 et 3. La programmation du Strang splitting est alors simple et directe.

Pour l'expérience numérique de la Fig. 9 nous choisissons $t_{\text{end}} = 400$, $h = t_{\text{end}}/5 = 80$ et nous considérons la méthode $\Phi^{[2]}_{h/2} \circ \Phi^{[1]}_h \circ \Phi^{[2]}_{h/2}$ où $\Phi^{[1]}_h$ et $\Phi^{[2]}_h$ représentent l'application du programme RADAU5 avec tolérance $Rtol = Atol = 10^{-5}$ au systèmes $\dot{y} = f^{[1]}(y)$ et $\dot{y} = f^{[2]}(y)$ respectivement. C.-à-d., pour le premier pas de la méthode, on applique d'abord le programme sur l'intervalle $[0, 40]$ à la réaction, ensuite sur $[0, 80]$ à la diffusion avec la nouvelle valeur initiale, enfin sur $[40, 80]$ à la réaction et ainsi de suite. Dans la Fig. 9, l'approximation numérique obtenue par $\Phi^{[2]}_h$ est colorée en bleu et celle de $\Phi^{[2]}_h$ en noir. Les valeurs initiales pour chaque intégration sont indiquées par un symbole plus grand.

Pour l'exemple considéré, l'influence de la diffusion est marginale (la difference entre les solutions exactes avec ou sans diffusion est à l'instant $t = 400$ de la grandeur $0.66 \cdot 10^{-5}$ pour la norme ℓ_2). Avec l'expérience de la Fig. 9 (mais avec $Rtol = Atol \leq$

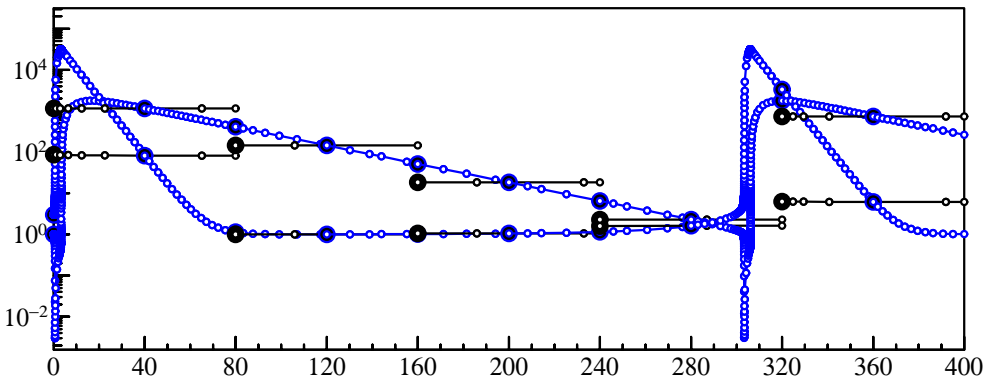

FIGURE 9: Strang splitting appliqué au système de réaction-diffusion avec $n = 100$; Sont dessinées les composantes b_i et c_i pour $x_i = 0.75$.

10^{-7}) l'erreur est réduite à $0.24 \cdot 10^{-6}$ bien que la solution soit corrigée par la diffusion seulement en cinq points de l'intervalle $[0, 400]$.

Plusieurs études théoriques, dont celles de [Spo00] et de [DM04], indiquent qu'il est préférable de choisir le Strang splitting qui se termine avec un demi-pas correspondant au flot de l'équation la plus raide.

3.3 Méthodes implicites-explicites (IMEX)

Comme dans le §3.2 nous considérons un système

$$\dot{y} = f(y) + g(y)$$

où le champs de vecteur est partitionné. La première partie $\dot{y} = f(y)$ est supposée être non raide, et la deuxième $\dot{y} = g(y)$ raide. L'idée est d'appliquer une méthode implicite (et A-stable) à la partie raide, mais une méthode explicite à la partie non raide.

3.3.1 Méthodes implicites-explicites de Runge–Kutta

Prenons les coefficients de deux méthodes de Runge–Kutta (une explicite et l'autre diagonalement implicite) et considérons le schéma

$$\begin{aligned} Y_i &= y_n + h\sum_{j=1}^{i-1} \widetilde{a}_{ij} f(Y_j) + h\sum_{j=1}^{i} a_{ij} g(Y_j) \qquad i = 1, \dots, s \\ y_{n+1} &= y_n + h\sum_{i=1}^{s} \widetilde{b}_i f(Y_i) + h\sum_{i=1}^{s} b_i g(Y_i). \end{aligned}$$

Chaque étage nécessite seulement une évaluation de $f(y)$ et la résolution d'un système non linéaire du type $Y_i = \eta_i + ha_{ii}g(Y_i)$ avec η_i donné, si $a_{ii} \neq 0$. Initialement [Gri78] ces méthodes ont été étudiées sous une forme équivalente où le vecteur y, et non le champ de vecteurs, est partitionné en raide et non raide. Des réferences plus récentes sont [ARS97], [PR01] et [KC03].

Pour que Y_i soit une approximation consistante de la solution $y(t_n + c_i h)$, nous supposons que

$$\sum_{j=1}^{i-1} \widetilde{a}_{ij} = \sum_{j=1}^{i} a_{ij} = c_i \qquad \text{pour} \quad i = 1, \dots, s.$$

Ceci nécessite que $a_{11} = c_1 = 0$, de sorte que le premier étage de la méthode soit explicite. Sans cette condition il est beaucoup plus laborieux de construire des méthodes avec un ordre élevé.

Conditions d'ordre. Pour l'ordre 2, seulement les conditions de quadrature

$$\sum_{i=1}^{s} \widetilde{b}_i = \sum_{i=1}^{s} b_i = 1, \qquad \sum_{i=1}^{s} \widetilde{b}_i c_i = \sum_{i=1}^{s} b_i c_i = \frac{1}{2}$$

doivent être satisfaites. Pour avoir l'ordre 3, il faut en plus que

$$\sum_{i=1}^{s} \widetilde{b}_i c_i^2 = \sum_{i=1}^{s} b_i c_i^2 = \frac{1}{3}, \qquad \sum_{i,j=1}^{s} \widetilde{b}_i \widetilde{a}_{ij} c_j = \sum_{i,j=1}^{s} \widetilde{b}_i a_{ij} c_j = \sum_{i,j=1}^{s} b_i \widetilde{a}_{ij} c_j = \sum_{i,j=1}^{s} b_i a_{ij} c_j = \frac{1}{6}.$$

On remarque qu'il ne suffit pas que chacune des méthodes pour $f(y)$ et $g(y)$ soit d'ordre 3, des conditions de couplage doivent aussi être satisfaites. Une théorie générale pour les conditions d'un ordre arbitraire est donnée dans [HNW93, §II.15].

Fonction de stabilité de la méthode implicite. La fonction de stabilité est donnée par $R(z) = 1 + b^T(I - zA)^{-1}\mathbb{1}$, où $b = (b_i)_{i=1}^s$, $A = (a_{ij})_{i,j=1}^s$ et $\mathbb{1} = (1, \ldots, 1)^T$. Elle n'est pas nécessairement bornée car A est singulière. Si $a_{11} = 0$ mais $a_{ii} > 0$ pour $i \geq 2$, son développement de Laurent autour du point infini est

$$R(z) = \alpha_1 z + \alpha_0 + \mathcal{O}(z^{-1}) \qquad \text{avec} \quad \alpha_1 = b_1 - \sum_{i,j=2}^{s} b_i w_{ij} a_{j1},$$

où la matrice $(w_{ij})_{i,j=2}^s$ est l'inverse de la matrice triangulaire $(a_{ij})_{i,j=2}^s$. Pour satisfaire $\alpha_1 = 0$, une de deux conditions suivantes suffit:

$$\text{(a)} \quad a_{si} = b_i \quad \text{pour} \quad i = 1, \ldots, s,$$

$$\text{(b)} \quad b_1 = 0 \quad \text{et} \quad a_{i1} = 0 \quad \text{pour} \quad i = 1, \ldots, s.$$

Si les deux conditions, (a) et (b), sont satisfaites on a $\alpha_1 = \alpha_0 = 0$ et la fonction de stabilité de la méthode satisfait $R(\infty) = 0$.

Précision pour des problèmes très raides ("stiff accuracy"). Pour des équations différentielles très raides (par exemple des problèmes à perturbation singulière) les étages internes sont de bonnes approximations de la solution $y(t_n + c_i h)$ dans le cas (b) ci-dessus. On peut obtenir la même propriété pour l'approximation numérique y_{n+1} en imposant la condition

$$\widetilde{b}_i = \widetilde{a}_{si}, \quad b_i = a_{si} \qquad \text{pour} \quad i = 1, \ldots, s,$$

ce qui implique que $y_{n+1} = Y_s$ est égal au dernier étage interne. Avec ce choix on a $\widetilde{b}_s = 0$, et par conséquent seulement $s - 1$ évaluations de la fonction $f(y)$ à effectuer à chaque pas d'intégration.

Un exemple d'une méthode implicite-explicite de Runge–Kutta qui satisfait toutes les conditions de ce paragraphe et qui est d'ordre 3 est donnée dans le tableau 6. Les coefficients de la méthode explicite sont à gauche, ceux de la méthode implicite à droite. Une collection de méthodes implicites-explicites de Runge–Kutta avec des méthodes emboîtées (pour une estimation de l'erreur et donc pour une programmation à pas variables) est donnée dans [KC03].

Cette classe de méthodes est souvent utilisée pour des semi-discrétisations d'équations aux dérivées partielles. Par exemple, des systèmes avec convection (non-raide) et diffusion (raide), ou des systèmes avec réaction (très raide) et convection-diffusion (moins raide).

Tableau 6: Coefficients d'une méthode IMEX d'ordre 3, [ARS97].

0	0	0	0	0	0
1/2	1/2	0	0	0	0
2/3	11/18	1/18	0	0	0
1/2	5/6	−5/6	1/2	0	0
1	1/4	7/4	3/4	−7/4	0
	1/4	7/4	3/4	−7/4	0

0	0	0	0	0	0
1/2	0	1/2	0	0	0
2/3	0	1/6	1/2	0	0
1/2	0	−1/2	1/2	1/2	0
1	0	3/2	−3/2	1/2	1/2
	0	3/2	−3/2	1/2	1/2

3.3.2 Méthodes multipas implicites-explicites

L'idée de méthodes implicites-explicites ne se restreint pas aux méthodes de Runge–Kutta. On peut également considérer des méthodes du type

$$\sum_{i=0}^{k} \alpha_i y_{n+i} = h \sum_{i=0}^{k-1} \widetilde{\beta}_i f(y_{n+i}) + h \sum_{i=0}^{k} \beta_i g(y_{n+i}).$$

Une seule évaluation de $f(y)$ par pas est nécessaire, ainsi que la résolution d'un système de la forme $y_{n+k} = \eta_n + h\gamma_k g(y_{n+k})$ avec $\gamma_k = \beta_k/\alpha_k$ et η_k donnés. Cette classe de méthodes est introduite dans [Cro80]. Pour l'étude de ces méthodes nous référons aux publications [ARW95], [ACM99] et [FHV97].

3.4 Equations différentielles-algébriques

Dans l'équation de Curtiss–Hirschfelder (§1.2), les solutions s'approchent rapidement d'une variété qui est proche de $y = \arctan(20\,t)$. Le même phénomène, mais dans une situation plus réaliste et dans une dimension supérieure, est caractéristique des problèmes à perturbation singulière

$$\dot{y} = f(y,z), \qquad \varepsilon\dot{z} = g(y,z),$$

où $0 < \varepsilon \ll 1$ est très petit et les valeurs propres de $\frac{\partial g}{\partial z}(y,z)$ possèdent une partie réelle négative le long de la solution exacte (pour garantir la stabilité). Les solutions s'approchent rapidement d'une variété qui est ε-proche de $g(y,z) = 0$.

Indice 1. En posant $\varepsilon = 0$ dans un problème à perturbation singulière on obtient une équation différentielle-algébrique (EDA)

$$\dot{y} = f(y,z), \qquad 0 = g(y,z),$$

c.-à-d., un système où une équation différentielle est couplée avec une équation algébrique. Si la matrice jacobienne $\frac{\partial g}{\partial z}(y,z)$ est inversible dans un voisinage de la solution,

on parle d'une EDA d'indice 1. Formellement, on peut résoudre l'équation $g(y,z)=0$ par rapport à z pour obtenir $z=G(y)$. Ainsi la dynamique du problème est complètement décrite par l'équation différentielle (non raide) $\dot{y}=f(y,G(y))$.

Indice 2. Si la matrice $\frac{\partial g}{\partial z}(y,z)$ n'est pas inversible, la situation devient beaucoup plus complexe. Considérons par exemple

$$\dot{y}=f(y,z), \qquad 0=g(y),$$

où z représente un contrôle qui permet de maintenir la solution $y(t)$ sur la variété $g(y)=0$. En dérivant cette relation algébrique par rapport au temps, on obtient une restriction supplémentaire $g'(y)f(y,z)=0$. Si $g'(y)\frac{\partial f}{\partial z}(y,z)$ est inversible, on peut de nouveau exprimer z en fonction de y et on obtient une équation différentielle pour y. Dans cette situation on parle d'une EDA d'indice 2.

Résolution numérique par des méthodes de Runge–Kutta. Appliquée à un problème à perturbation singulière, une méthode de Runge–Kutta implicite donne

$$\begin{aligned} Y_i - y_n &= h\sum_{j=1}^{s} a_{ij}\, f(Y_j,Z_j) & \qquad \varepsilon(Z_i-z_n) &= h\sum_{j=1}^{s} a_{ij}\, g(Y_j,Z_j) \\ y_{n+1}-y_n &= h\sum_{i=1}^{s} b_i\, f(Y_i,Z_i) & \qquad \varepsilon(z_{i+1}-z_n) &= h\sum_{i=1}^{s} b_i\, g(Y_i,Z_i). \end{aligned}$$

Dans le cas limite $\varepsilon=0$, si la matrice de Runge–Kutta (a_{ij}) est inversible, on a $g(Y_j,Z_j)=0$ pour $j=1,\dots,s$ et les étages internes sont sur la même sous-variété que la solution exacte. Pour avoir cette propriété aussi pour l'approximation numérique y_{n+1} nous supposons que $b_i=a_{si}$ pour $i=1,\dots,s$ de sorte que $y_{n+1}=Y_s$.

Pour un problème d'indice 1, cette approche est équivalente à une application de la méthode au système $\dot{y}=f(y,G(y))$. Par conséquence, l'erreur globale satisfait $y_n-y(t_n)=\mathcal{O}(h^p)$ et $z_n-z(t_n)=G(y_n)-G(y(t_n))=\mathcal{O}(h^p)$ où p est l'ordre (classique) de la méthode. Rappelons que pour les méthodes Radau IIA du §2.1 l'ordre est $p=2s-1$.

Pour un problème d'indice 2 (où l'équation algébrique ne dépend que de y) on peut démontrer que le système non linéaire de la méthode de Runge–Kutta possède une solution si les valeurs $g(y_n)$ et $g'(y_n)f(y_n,z_n)$ sont suffisamment petites. L'erreur globale satisfait $y_n-y(t_n)=\mathcal{O}(h^p)$ comme pour des équations différentielles non raides, mais on a seulement $z_n-z(t_n)=\mathcal{O}(h^q)$ avec un $q\le p$ (phénomène de réduction d'ordre; on peut appeler q l'ordre interne de la méthode). Par exemple, pour les méthodes Radau IIA on a $p=2s-1$ et $q=s$. Plus de détails se trouvent dans [HLR89] et [HW96, §VI.3].

Considérons de nouveau un problème à perturbation singulière qui est un cas particulier très important des équations raides, et supposons que les valeurs initiales soient proches d'une solution lisse. Appliquons une méthode de Runge–Kutta d'ordre p,

d'ordre interne q, qui est L-stable, qui possède une matrice (a_{ij}) inversible et qui satisfait $a_{si} = b_i$ pour tout i (par exemple, Radau IIA). En développant la solution exacte et la solution numérique en série de puissances de ε, et en utilisant les résultats de convergence pour les équations différentielles-algébriques, on peut démontrer que

$$y_n - y(t_n) = \mathcal{O}(h^p) + \mathcal{O}(\varepsilon^2 h^q), \qquad z_n - z(t_n) = \mathcal{O}(h^p) + \mathcal{O}(\varepsilon h^q).$$

Ce résultat montre la continuité de l'erreur pour $\varepsilon \to 0$. Ainsi, ces méthodes ont le même comportement pour $\varepsilon = 10^{-4}$, $\varepsilon = 10^{-12}$ ou $\varepsilon = 0$.

4 Appendice : Programmes informatiques

Pour la résolution des équations différentielles raides, il existe d'excellents programmes du domaine publique qui peuvent être téléchargés sur internet.

- `http://www.unige.ch/~hairer/software.html`

  RADAU5 est basé sur une méthode de Runge–Kutta implicite (Radau IIA, §2.1.1) d'ordre 5; RADAU est une version avec ordre variable (ordres 5, 9, et 13); SEULEX est la méthode d'extrapolation basée sur la méthode d'Euler linéairement implicite, discutée au §2.3; ROCK est une méthode de Runge–Kutta explicite avec un grand intervalle de stabilité (§3.1.2).

- `http://www.netlib.org/ode/index.html`

  DASSL est un programme célèbre de L. Petzold basé sur les méthodes BDF [BCP96]; RKC est la méthode explicite discutée dans le §3.1.1.

- `http://www.llnl.gov/casc/odepack/download/lsode_agree.html`

  LSODE est une programmation des méthodes multipas BDF.

- `http://www.zib.de/Numerik/numsoft/CodeLib/ivpode.en.html`

  EULSIM est la méthode d'extrapolation basée sur la méthode d'Euler linéairement implicite, discutée au §2.3. LIMEX est une extension aux problèmes différentiels-algébriques.

- `http://www.ma.ic.ac.uk/~jcash/IVP_software/readme.html`

  MEBDF est synonyme de "Modified Extended BDF" [CC92].

- `http://pitagora.dm.uniba.it/~mazzia/ode/gam.html`

  GAM est une méthode de Runge–Kutta implicite, basée sur l'idée de "boundary value methods" [IM98].

Plusieurs de ces programmes informatiques sont écrits pour des équations différentielles de la forme

$$M\,\dot{y} = f(t, y), \qquad y(t_0) = y_0$$

où la matrice M peut être singulière pour permettre d'inclure des relations algébriques. Le vecteur $f(t_0, y_0)$ doit être dans l'image de l'application linéaire M. D'autres programmes sont écrit pour des problèmes encore plus généraux

$$F(\dot{y}, y, t) = 0, \qquad y(t_0) = y_0, \quad \dot{y}(t_0) = \dot{y}_0$$

avec des valeurs initiales consistantes, c.-à-d., $F(\dot{y}_0, y_0, t_0) = 0$. Remarquons qu'avec l'introduction d'une nouvelle variable pour la dérivée $z = \dot{y}$, le problème devient équivalent au système

$$\dot{y} = z, \qquad 0 = F(z, y, t),$$

qui est de nouveau sous la forme $M\dot{Y} = G(Y, t)$ où le vecteur Y est composé par y et z.

Sur la page internet `http://pitagora.dm.uniba.it/~testset/` on peut trouver la description de plusieurs programmes pour des problèmes raides et différentiels-algébriques, une collection importante d'exemples intéressants et aussi des comparaisons numériques.

References

[Abd02] A. Abdulle. Fourth order Chebyshev methods with recurrence relation. *SIAM J. Sci. Comput.*, 23(6):2041–2054 (electronic), 2002.

[ACM99] G. Akrivis, M. Crouzeix, and C. Makridakis. Implicit-explicit multistep methods for quasilinear parabolic equations. *Numer. Math.*, 82(4):521–541, 1999.

[Aik85] R. C. Aiken, editor. *Stiff computation*. Oxford University Press, New York, 1985.

[AM01] A. Abdulle and A. A. Medovikov. Second order Chebyshev methods based on orthogonal polynomials. *Numer. Math.*, 90(1):1–18, 2001.

[ARS97] U. M. Ascher, S. J. Ruuth, and R. J. Spiteri. Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Appl. Numer. Math.*, 25(2-3):151–167, 1997. Special issue on time integration (Amsterdam, 1996).

[ARW95] U. M. Ascher, S. J. Ruuth, and B. T. R. Wetton. Implicit-explicit methods for time-dependent partial differential equations. *SIAM J. Numer. Anal.*, 32(3):797–823, 1995.

[BCP96] K. E. Brenan, S. L. Campbell, and L. R. Petzold. *Numerical solution of initial-value problems in differential-algebraic equations*, volume 14 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996. Revised and corrected reprint of the 1989 original.

[CC92] J. R. Cash and S. Considine. An MEBDF code for stiff initial value problems. *ACM Trans. Math. Software*, 18(2):142–155, 1992.

[CH52] C. F. Curtiss and J. O. Hirschfelder. Integration of stiff equations. *Proc. Nat. Acad. Sci. U. S. A.*, 38:235–243, 1952.

[Cro80] M. Crouzeix. Une méthode multipas implicite-explicite pour l'approximation des équations d'évolution paraboliques. *Numer. Math.*, 35(3):257–276, 1980.

[Dah63] G. Dahlquist. A special stability problem for linear multistep methods. *BIT*, 3:27–43, 1963.

[Deu85] P. Deuflhard. Recent progress in extrapolation methods for ordinary differential equations. *SIAM Rev.*, 27(4):505–535, 1985.

[DM04] S. Descombes and M. Massot. Operator splitting for nonlinear reaction-diffusion systems with an entropic structure: singular perturbation and order reduction. *Numer. Math.*, 97(4):667–698, 2004.

[DV84] K. Dekker and J. G. Verwer. *Stability of Runge-Kutta methods for stiff nonlinear differential equations*, volume 2 of *CWI Monographs*. North-Holland Publishing Co., Amsterdam, 1984.

[Ehl69] B. L. Ehle. On Padé approximations to the exponential function and a-stable methods for the numerical solution of initial value problems. Technical Report CSRR 2010, Dept. AACS Univ. of Waterloo Ontario Canada., 1969.

[FHV97] J. Frank, W. Hundsdorfer, and J. G. Verwer. On the stability of implicit-explicit linear multistep methods. *Appl. Numer. Math.*, 25(2-3):193–205, 1997. Special issue on time integration (Amsterdam, 1996).

[FN74] J. Field and R. M. Noyes. Oscillations in chemical systems. IV: Limit cycle behaviour in a model of a real chemical reaction. *J. Chem. Phys.*, 60:1877–1884, 1974.

[GL61] A. Guillou and B. Lago. Domaine de stabilité associé aux formules d'intégration numérique d'équations différentielles, à pas séparés et à pas liés. Recherche de formules à grand rayon de stabilité. *1er Congr. Assoc. Fran. Calcul, AFCAL, Grenoble, Sept. 1960*, pages 43–56, 1961.

[Gri78] E. Griepentrog. Gemischte Runge-Kutta-Verfahren für steife Systeme. In *Berichte der Tagung "Numerische Behandlung Gewöhnlicher Differentialgleichungen" (Humboldt Univ., Berlin, 1978)*, volume 11 of *Seminarberichte*, pages 19–29. Humboldt Univ. Berlin, 1978.

[GS69] A. Guillou and J. L. Soulé. La résolution numérique des problèmes différentiels aux conditions initiales par des méthodes de collocation. *Rev. Française Informat. Recherche Opérationnelle*, 3(Ser. R-3):17–44, 1969.

[HLR89] E. Hairer, C. Lubich, and M. Roche. The numerical solution of differential-algebraic systems by Runge-Kutta methods. In *Lecture Notes in Math. 1409*. Springer-Verlag, 1989.

[HNW93] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I. Nonstiff Problems*. Springer Series in Computational Mathematics 8. Springer, Berlin, 2 edition, 1993.

[HS80] P. J. van der Houwen and B. P. Sommeijer. On the internal stability of explicit, m-stage, Runge–Kutta methods for large m-values. *Z. Angew. Math. Mech.*, 60:479–485, 1980.

[HV03] W. Hundsdorfer and J. Verwer. *Numerical solution of time-dependent advection-diffusion-reaction equations*, volume 33 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2003.

[HW96] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*. Springer Series in Computational Mathematics 14. Springer-Verlag, Berlin, 2 edition, 1996.

[IM98] F. Iavernaro and F. Mazzia. Solving ordinary differential equations by generalized Adams methods: properties and implementation techniques. *Appl. Numer.*

Math., 28(2-4):107–126, 1998. Eighth Conference on the Numerical Treatment of Differential Equations (Alexisbad, 1997).

[JN81] R. Jeltsch and O. Nevanlinna. Stability of explicit time discretizations for solving initial value problems. *Numer. Math.*, 37(1):61–91, 1981.

[KC03] C. A. Kennedy and M. H. Carpenter. Additive Runge-Kutta schemes for convection-diffusion-reaction equations. *Appl. Numer. Math.*, 44(1-2):139–181, 2003.

[Leb89] V. I. Lebedev. Explicit difference schemes with time-variable steps for solving stiff systems of equations. *Soviet J. Numer. Anal. Math. Modelling*, 4(2):111–135, 1989.

[Mar68] G. Marchuk. Some applications of splitting-up methods to the solution of mathematical physics problems. *Aplikace Matematiky*, 13:103–132, 1968.

[Mir81] W. L. Miranker. *Numerical methods for stiff equations and singular perturbation problems*, volume 5 of *Mathematics and its Applications*. D. Reidel Publishing Co., Dordrecht, 1981.

[PR74] A. Prothero and A. Robinson. On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations. *Math. Comp.*, 28:145–162, 1974.

[PR01] L. Pareschi and G. Russo. Implicit-explicit Runge-Kutta schemes for stiff systems of differential equations. In *Recent trends in numerical analysis*, volume 3 of *Adv. Theory Comput. Math.*, pages 269–288. Nova Sci. Publ., Huntington, NY, 2001.

[Spo00] B. Sportisse. An analysis of operator splitting techniques in the stiff case. *J. Comput. Phys.*, 161(1):140–168, 2000.

[SSV98] B. P. Sommeijer, L. F. Shampine, and J. G. Verwer. RKC: an explicit solver for parabolic PDEs. *J. Comput. Appl. Math.*, 88(2):315–326, 1998.

[Str68] G. Strang. On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.*, 5:506–517, 1968.

[SV80] B. P. Sommeijer and J. G. Verwer. *A performance evaluation of a class of Runge-Kutta-Chebyshev methods for solving semidiscrete parabolic differential equations*. Afdeling Numerieke Wiskunde [Department of Numerical Mathematics], 91. Mathematisch Centrum, Amsterdam, 1980.

[Tem68] R. Temam. Sur la stabilité et la convergence de la méthode des pas fractionnaires. *Ann. Mat. Pura Appl. (4)*, 79:191–379, 1968.

[WHN78] G. Wanner, E. Hairer, and S. P. Nørsett. Order stars and stability theorems. *BIT*, 18(4):475–489, 1978.

[Wil74] R. A. Willoughby, editor. *Stiff differential systems*. Plenum Press, New York, 1974. The IBM Research Symposia Series.

[Wri70] K. Wright. Some relationships between implicit Runge-Kutta collocation and Lanczos τ methods and their stability properties. *BIT*, 10:217–227, 1970.