Méthodes numériques et éléments de programmation

Guy Munhoven

Institut d'Astrophysique et de Géophysique (Bât. B5c)
Bureau 0/13
eMail: Guy.Munhoven@ulg.ac.be
Tél.: 04-3669771

22 septembre 2014

Méthodes numériques et éléments de programmation

Guy Munhoven

Rappels

Runge-Kutta

Stabilité absolue et A-Stabilité

Pas variables

Méth. multi-pas

Doublement

Ordres p, $p+1$

Problèmes raides

Stabilité : compléments

Plan du cours 2014-2015

Cours théoriques

16-09-2014 *Méthodes numériques pour équations différentielles ordinaires : introduction*

22-09-2014 Méthodes de Runge-Kutta ; Contrôle du pas

- Méthodes de Runge-Kutta
- Erreur et contrôle du pas
- Equations raides

22-09-2014 Fortran 95 : bases

4–5 cours Fortran 95 : suite et compléments

Méthodes obtenues par quadrature numérique

Methods Obtained by Numerical Quadrature

Problème à valeur initiale sur intervalle $[t_0, t_0 + T]$ fixé :

$$\frac{dy}{dt} = f(y), \quad \text{pour} \quad t \in [t_0, t_0 + T], \quad \text{avec} \quad y(t_0) = y_0$$

Cinq étapes :

1. Choisir une grille de discrétisation
2. Intégrer l'équation différentielle entre deux points de la grille, p.ex., t_j et t_{j+1} :

$$\int_{t_j}^{t_{j+1}} y'(t_j)\, dt = \int_{t_j}^{t_{j+1}} f(y(t))\, dt.$$

Donc :

$$y(t_{j+1}) - y(t_j) = \int_{t_j}^{t_{j+1}} f(y(t))\, dt.$$

Méthodes obtenues par quadrature numérique

Methods Obtained by Numerical Quadrature

Cinq étapes (suite) :

3. Remplacer l'intégrale de f par une formule aux différences finies, comme par exemple, la formule des trapèzes

$$y(t_{j+1}) - y(t_j) = \frac{k}{2}[f(y(t_{j+1})) + f(y(t_j))] + O(k^3)$$

$O(k^3)$ est une fonction telle que $\lim_{k\to 0}(O(k^3)/k^3)$ est finie.

Cinq étapes (suite) :

4. Laisser tomber le terme $O(k^3)$ et remplacer $y(t_j)$ par y_j, etc. :

$$y_{j+1} = y_j + \frac{k}{2}(f_{j+1} + f_j)$$

Méthode des trapèzes, implicite

5. A-stabilité

Méthodes obtenues par quadrature numérique
Methods Obtained by Numerical Quadrature

Méthodes numériques et éléments de programmation

Guy Munhoven

Rappels
Runge-Kutta
Stabilité absolue et A-Stabilité
Pas variables
Méth. multi-pas
Doublement
Ordres p, $p+1$
Problèmes raides
Stabilité : compléments

Autres procédures pour quadrature numérique :

Approximation de $f(y(t))$ sur $[t_j, t_{j+1}]$ par un polynôme de degré $q \geq 0$ passant par

- $(t_{j-q}, f(y_{j-q})), \ldots, (t_j, f(y_j))$

  méthode d'Adams-Bashforth (explicite)
- $(t_{j-(q-1)}, f(y_{j-(q-1)})), \ldots, (t_{j+1}, f(y_{j+1}))$

  méthode d'Adams-Moulton (implicite)

Développement de méthodes d'ordre supérieur
Development of Higher Order Methods

Stratégies pour construire des méthodes d'ordre supérieur

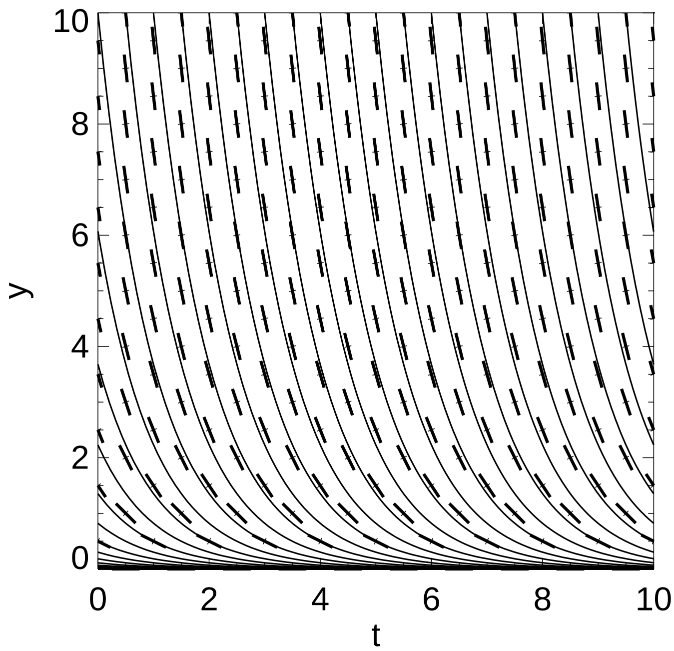
Méthodes multi-pas

- linéaires
- une seule évaluation du second membre par pas
- modification du pas d'intégration compliquée

Méthodes de Runge-Kutta

- méthodes généralement non-linéaires
- plusieurs évaluations du second membre pour effectuer un seul pas
- modification du pas d'intégration facile

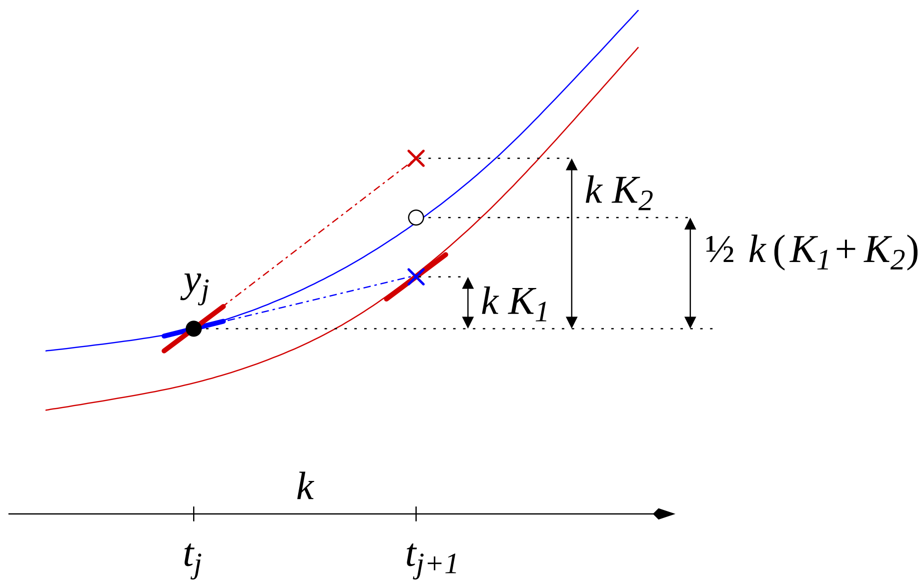
Méthodes de Runge-Kutta : préliminaires
Runge-Kutta Methods : Preliminaries



$$\frac{dy}{dt} = -y$$

Méthodes de Runge-Kutta : logique

Runge-Kutta Methods : Rationale



Méthodes de Runge-Kutta : logique

Runge-Kutta Methods : Rationale

Méthodes numériques et éléments de programmation

Guy Munhoven

Rappels
Runge-Kutta
Stabilité absolue et A-Stabilité
Pas variables
Méth. multi-pas
Doublement
Ordres p, $p+1$
Problèmes raides
Stabilité : compléments

Problème donné :

$$\frac{dy}{dt} = f(t,y), \quad \text{pour} \quad t \in [0,T], \quad \text{avec} \quad y(0) = y_0$$

Intégrer l'équation différentielle entre t_j et t_{j+1} $(t_{j+1} = t_j + k)$:

$$\int_{t_j}^{t_{j+1}} y'(t_j)\,dt = \int_{t_j}^{t_{j+1}} f(t,y(t))\,dt$$

Donc :

$$y(t_{j+1}) - y(t_j) \quad = \quad \int_{t_j}^{t_{j+1}} f(t,y(t))\,dt$$

Méthodes de Runge-Kutta : logique

Runge-Kutta Methods : Rationale

- Changement de variable $t = t_j + ck$, avec $k = t_{j+1} - t_j$

$$y(t_{j+1}) - y(t_j) = k \int_0^1 f(t_j + ck, y(t_j + ck))\, dc$$

- Remplacer l'intégrale par une formule de quadrature

$$y(t_{j+1}) \simeq y(t_j) + k \sum_{m=1}^{s} b_m f(t_j + c_m k, y(t_j + c_m k))$$

  où
  - s est le nombre d'*étapes* par pas
  - les c_m (à déterminer) définissent les noeuds où f est évalué
  - les b_m (à déterminer) pondèrent ces valeurs évaluées
- $y(t_j + c_m k)$ inconnus $\Rightarrow$ approximations

Méthodes de Runge-Kutta : logique

Runge-Kutta Methods : Rationale

Pour les méthodes de Runge-Kutta explicites

- premier noeud

  fixé en t_j, où f est connu $\Rightarrow c_1 = 0$
- noeuds $m = 2, \ldots, s$

  $y(t_j + c_m k)$ estimé à partir de $y(t_j)$, auquel une combinaison linéaire des $f(t_j + c_n k, y(t_j + c_n k))$ aux noeuds $n < m$ (déjà calculés) est ajoutée :

$$y(t_j + c_m k) \simeq y_j + k \sum_{n=1}^{m-1} a_{mn} f(t_j + c_n k, y(t_j + c_n k))$$

- a_{mn}, b_n, c_m $(m, n = 1, \ldots, s)$ déterminés de manière à ce que l'approximation soit compatible avec le développement en série de Taylor de la solution exacte

Pour la solution exacte :

$$y(t_{j+1}) = y(t_j) + ky'|_{t_j} + \frac{k^2}{2} y''|_{t_j} + O(k^3)$$

Or

$$\begin{aligned} y' &= f \\ y'' &= \frac{\partial f}{\partial t} + \frac{\partial f}{\partial y} y' = f_t + f_y f \end{aligned}$$

Ainsi

$$y(t_{j+1}) = y(t_j) + kf|_{t_j} + \frac{k^2}{2}(f_t + f_y f)|_{t_j} + O(k^3)$$

Méthodes de Runge-Kutta explicites à deux étapes
Two-stage Explicit Runge-Kutta Methods

Méthodes numériques et éléments de programmation

Guy Munhoven

Rappels

Runge-Kutta

Stabilité absolue et A-Stabilité

Pas variables

Méth. multi-pas

Doublement

Ordres p, $p+1$

Problèmes raides

Stabilité : compléments

Pour l'approximation au deuxième noeud :

$$\begin{aligned} & f(t_j + c_2 k, y(t_j + c_2 k)) \\ &\quad = f(t_j + c_2 k, y(t_j) + a_{21} k f(t_j, y_j)) \\ &\quad = f(t_j, y(t_j)) + c_2 k f_t|_{t_j} + a_{21} k (f_y f)|_{t_j} + O(k^2) \end{aligned}$$

Ainsi

$$\begin{aligned} y_{j+1} &= y(t_j) + kb_1 f(t_j, y(t_j)) + kb_2 f(t_j + c_2 k, y(t_j + c_2 k)) \\ &= y(t_j) + k(b_1 + b_2) f|_{t_j} \\ &\qquad + b_2 c_2 k^2 f_t|_{t_j} + b_2 a_{21} k^2 (f_y f)|_{t_j} + O(k^3) \end{aligned}$$

à comparer avec

$$y(t_{j+1}) = y(t_j) + kf|_{t_j} + \tfrac{k^2}{2} f_t|_{t_j} + \tfrac{k^2}{2}(f_y f)|_{t_j} + O(k^3)$$

Méthodes de Runge-Kutta explicites à deux étapes

Two-stage Explicit Runge-Kutta Methods

Conditions suffisantes de compatibilité :

$$b_1 + b_2 = 1 \qquad b_2 c_2 = \frac{1}{2} \qquad b_2 a_{21} = \frac{1}{2}$$

Choix particuliers :

$$c_2 = \frac{1}{2} \qquad\qquad c_2 = \frac{2}{3} \qquad\qquad c_2 = 1$$

$$\begin{array}{rr}
c_1 = 0 & b_1 = 0 \\
c_2 = \frac{1}{2} & b_2 = 1 \\
a_{11} = 0 & a_{12} = 0 \\
a_{21} = \frac{1}{2} & a_{22} = 0
\end{array}
\qquad
\begin{array}{rr}
c_1 = 0 & b_1 = \frac{1}{4} \\
c_2 = \frac{2}{3} & b_2 = \frac{3}{4} \\
a_{11} = 0 & a_{12} = 0 \\
a_{21} = \frac{2}{3} & a_{22} = 0
\end{array}
\qquad
\begin{array}{rr}
c_1 = 0 & b_1 = \frac{1}{2} \\
c_2 = 1 & b_2 = \frac{1}{2} \\
a_{11} = 0 & a_{12} = 0 \\
a_{21} = 1 & a_{22} = 0
\end{array}$$

Toutes d'ordre 2

Méthodes de Runge-Kutta

Runge-Kutta Methods

Méthodes numériques et éléments de programmation
Guy Munhoven
Rappels
Runge-Kutta
Stabilité absolue et A-Stabilité
Pas variables
Méth. multi-pas
Doublement
Ordres p, $p+1$
Problèmes raides
Stabilité : compléments

Une méthode de Runge-Kutta à s étapes s'écrit sous la forme

$$y_{j+1} = y_j + k\Phi(t_j, y_j, k; f).$$

$\Phi(t_j, y_j, k; f)$ est appelée *fonction d'incrément* et est définie par

$$\Phi(t_j, y_j, k; f) = \sum_{m=1}^{s} b_m K_m$$

$$K_m = f(t_j + c_m k, y_j + k \sum_{n=1}^{s} a_{mn} K_n), \quad m = 1, \dots s.$$

Méthodes de Runge-Kutta
Runge-Kutta Methods

Une méthode de Runge-Kutta à s étapes est complètement définie par ses coefficients a_{mn}, b_m et c_m $(m, n = 1, \ldots, s)$. Ils sont communément présentés dans le tableau de *Butcher*

$$\begin{array}{c|cccccc} c_1 & a_{11} & a_{12} & \cdots & a_{1,s-1} & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2,s-1} & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{s-1} & a_{s-1,1} & a_{s-1,2} & \cdots & a_{s-1,s-1} & a_{s-1,s} \\ c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_{s-1} & b_s \end{array} \quad \text{ou} \quad \begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b}^T \end{array}$$

Méthodes de Runge-Kutta : exemples
Runge-Kutta Methods : Examples

Méthodes numériques et éléments de programmation

Guy Munhoven

Rappels
Runge-Kutta
Stabilité absolue et A-Stabilité
Pas variables
Méth. multi-pas
Doublement
Ordres p, $p+1$
Problèmes raides
Stabilité : compléments

Méthode de Heun :

$$\begin{aligned} K_1 &= f(t_j, y_j) \\ K_2 &= f(t_j + k, y_j + kK_1) \\ \Phi(t_j, y_j, k; f) &= \tfrac{1}{2}(K_1 + K_2) \end{aligned}$$

$$\begin{array}{c|cc} 0 & \cdot & \cdot \\ 1 & 1 & \cdot \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

Méthode de Runge-Kutta (classique) :

$$
\begin{aligned}
K_1 &= f(t_j, y_j)\\
K_2 &= f(t_j + \tfrac{k}{2}, y_j + \tfrac{k}{2}K_1)\\
K_3 &= f(t_j + \tfrac{k}{2}, y_j + \tfrac{k}{2}K_2)\\
K_4 &= f(t_j + k, y_j + kK_3)\\
\Phi(t_j, y_j, k; f) &= \tfrac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)
\end{aligned}
$$

$$
\begin{array}{c|cccc}
0 & \cdot & \cdot & \cdot & \cdot\\
\frac{1}{2} & \frac{1}{2} & \cdot & \cdot & \cdot\\
\frac{1}{2} & \cdot & \frac{1}{2} & \cdot & \cdot\\
1 & \cdot & \cdot & 1 & \cdot\\
\hline
 & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6}
\end{array}
$$

Méthodes de Runge-Kutta : exemples
Runge-Kutta Methods : Examples

Méthodes numériques et éléments de programmation

Guy Munhoven

Rappels
Runge-Kutta
Stabilité absolue et A-Stabilité
Pas variables
Méth. multi-pas
Doublement
Ordres p, $p+1$
Problèmes raides
Stabilité : compléments

Méthode d'Euler explicite :

$$
\begin{aligned}
K_1 &= f(t_j, y_j)\\
\Phi(t_j, y_j, k; f) &= K_1
\end{aligned}
$$

$$
\begin{array}{c|c}
0 & \cdot\\
\hline
 & 1
\end{array}
$$

Méthode d'Euler implicite : une méthode de Runge-Kutta ?

$$\begin{aligned} y_{j+1} &= y_j + kf(t_{j+1}, y_{j+1}) \\ s &= 1 \\ \Phi(t_j, y_j, k; f) &= K_1 \\ K_1 &= f(t_{j+1}, y_{j+1}) \\ &= f(t_j + k, y_j + kK_1) \end{aligned}$$

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$$

Méthode des trapèzes ?

$$\begin{aligned} y_{j+1} &= y_j + k(\tfrac{1}{2}f(t_j, y_j) + \tfrac{1}{2}f(t_{j+1}, y_{j+1})) \\ s &= 2 \\ \Phi(t_j, y_j, k; f) &= \tfrac{1}{2}K_1 + \tfrac{1}{2}K_2 \\ K_1 &= f(t_j, y_j) \\ K_2 &= f(t_{j+1}, y_{j+1}) \\ &= f(t_j + k, y_j + k(\tfrac{1}{2}K_1 + \tfrac{1}{2}K_2)) \end{aligned}$$

$$\begin{array}{c|cc} 0 & \cdot & \cdot \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

Méthodes de Runge-Kutta : types

Runge-Kutta Methods : Types

- Méthodes explicites (ERK – *Explicit Runge-Kutta*)
  - tableau (a_{mn}) *strictement* triangulaire inférieur
  - tous les $K_m = y(t_j + c_m k)$ peuvent être calculés de manière explicite, les uns après les autres
- Méthodes implicites (IRK – *Implicit Runge-Kutta*)
  - $a_{mn} \neq 0$, pour $m \leq n$
  - les K_m doivent évt. être calculés tous ensemble
  - cas particulier : DIRK (*Diagonally Implicit Runge-Kutta*)

méthodes où $a_{mn} = 0$ pour $m < n$

$\Rightarrow$ (a_{mn}) triangulaire inférieur

$\Rightarrow$ les K_m peuvent encore être calculés à tour de rôle

$$K_m = y_j + k \sum_{n=1}^{m-1} a_{mn} f(t_j + c_n k, K_n) + k a_{mm} f(t_j + c_m k, K_m)$$

Méthodes de Runge-Kutta : ordre et étapes

Runge-Kutta Methods : Order and Stages

Méthodes numériques et éléments de programmation

Guy Munhoven

Rappels

Runge-Kutta

Stabilité absolue et A-Stabilité

Pas variables

Méth. multi-pas

Doublement

Ordres p, $p+1$

Problèmes raides

Stabilité : compléments

- ERK

Ordre vs. nombre minimal d'étapes requises pour ERK

Ordre	1	2	3	4	5	6	7	8
$s_{\min}$	1	2	3	4	6	7	9	11

- IRK

Théorême (Butcher)

Pour tout entier $s \geq 1$, il existe une et une seule méthode IRK d'ordre $2s$ à s étapes.

La seule méthode IRK à s étapes d'ordre $2s$ est la méthode de Gauss-Legendre, obtenue à partir des formules de quadrature de Gauss-Legendre par un procédé de *collocation* :

- déterminer un polynôme $u(t)$ sur $[t_j, t_{j+1}]$, dépendant de s paramètres *distincts* $c_1, \ldots, c_s$ (dits *paramètres de collocation*), tel que, pour $m = 1, \ldots, s$

$$\begin{aligned} u(t_j) &= y_j \\ u'(t_j + c_m k) &= f(t_j + c_m k, u(t_j + c_m k)). \end{aligned}$$

- adopter $y_{n+1} = u(t_{n+1})$

Pour Runge-Kutta Gauss-Legendre :

- u déterminé à partir du polynôme de Legendre d'ordre $2s$, défini par convention pour $c \in]-1, 1[$, transformé ici de manière linéaire pour $c \in]0, 1[$ et noté $q(c)$
- les c_m sont les racines (*distinctes une à une !*) de $q(c)$:

$$q(c) = \prod_{m=1}^{s} (c - c_m)$$

- en définissant $q_r(c) = q(c)/(c - c_r)$, on a, pour $m, n = 1, \ldots, s$:

$$b_m = \int_0^1 \frac{q_m(c)}{q_m(c_m)} dc \quad \text{et} \quad a_{mn} = \int_0^{c_m} \frac{q_n(c)}{q_n(c_n)} dc.$$

RK Gauss-Legendre

RK-GL2

$$\begin{array}{c|c} \frac{1}{2} & \frac{1}{2} \\ \hline & 1 \end{array}$$

RK-GL6

$$\begin{array}{c|ccc} \frac{1}{2}-\frac{\sqrt{15}}{10} & \frac{5}{36} & \frac{2}{9}-\frac{\sqrt{15}}{15} & \frac{5}{36}-\frac{\sqrt{15}}{30} \\ \frac{1}{2} & \frac{5}{36}+\frac{\sqrt{15}}{24} & \frac{2}{9} & \frac{5}{36}-\frac{\sqrt{15}}{24} \\ \frac{1}{2}+\frac{\sqrt{15}}{10} & \frac{5}{36}+\frac{\sqrt{15}}{30} & \frac{2}{9}+\frac{\sqrt{15}}{15} & \frac{5}{36} \\ \hline & \frac{5}{18} & \frac{4}{9} & \frac{5}{18} \end{array}$$

RK-GL4

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

Stabilité absolue et A-stabilité en général

Absolute Stability and A-stability in General

- Problème test

$$y'(t) = \lambda y, \quad y(0) = 1 \quad t > 0, \lambda \in \mathbb{C}$$

- Solution : $y(t) = e^{\lambda t}$
- $\lim_{t\to+\infty} |y(t)| = 0$ si $\Re(\lambda) < 0$

Stabilité absolue et A-stabilité en général
Absolute Stability and A-stability in General

Définition

Une méthode numérique pour l'approximation du problème test est absolument stable si

$$|y_j| \to 0 \text{ lorsque } t_j \to +\infty$$

La région de stabilité absolue est le sous-ensemble du plan complexe

$$\mathscr{A} = \{z = k\lambda \in \mathbb{C} | \lim_{t_j \to +\infty} |y_j| = 0\}$$

Une méthode est A-stable si $\mathbb{C}^- \subseteq \mathscr{A}$, *où* $\mathbb{C}^- = \{z \in \mathbb{C} | \Re(z) < 0\}$.

A-stabilité des méthodes de Runge-Kutta
A-stability of Runge-Kutta Methods

Méthodes numériques et éléments de programmation

Guy Munhoven

Rappels
Runge-Kutta
Stabilité absolue et A-Stabilité
Pas variables
Méth. multi-pas
Doublement
Ordres p, $p+1$
Problèmes raides
Stabilité : compléments

- Méthode de Runge-Kutta appliquée au problème test

$$y_{j+1} = y_j + k \sum_{m=1}^{s} b_m K_m, \quad K_m = \lambda \left(y_j + k \sum_{n=1}^{s} a_{mn} K_n \right)$$

- Avec $\mathbf{K} = [K_1, \ldots, K_s]^\mathsf{T}$ et $\mathbf{1} = [1, \ldots, 1]^\mathsf{T}$, cette récurrence peut être ré-écrite sous forme vectorielle

$$y_{j+1} = y_j + k\mathbf{b}^\mathsf{T}\mathbf{K}, \quad \mathbf{K} = \lambda(y_j\mathbf{1} + k\mathbf{A}\mathbf{K}).$$

- Ainsi, $\mathbf{K} = \lambda(I - k\lambda\mathbf{A})^{-1}\mathbf{1}y_j$

A-stabilité des méthodes de Runge-Kutta
A-stability of Runge-Kutta Methods

- Une itération d'une méthode de Runge-Kutta peut donc se mettre sous la forme

$$y_{j+1} = (1 + k\lambda \mathbf{b}^{\mathsf{T}}(I - k\lambda \mathbf{A})^{-1}\mathbf{1})y_j = R(k\lambda)y_j$$

- $R(k\lambda)$ est appelé *fonction de stabilité*
- Méthode absolument stable si $|R(k\lambda)| < 1$
- Région de stabilité d'une méthode de Runge-Kutta

$$\mathscr{A} = \{z = k\lambda \in \mathbb{C} \text{ tels que } |R(k\lambda)| < 1\}$$

A-stabilité des méthodes ERK
A-stability of ERK Methods

Méthodes numériques et éléments de programmation

Guy Munhoven

Rappels

Runge-Kutta

Stabilité absolue et A-Stabilité

Pas variables

Méth. multi-pas

Doublement

Ordres p, $p+1$

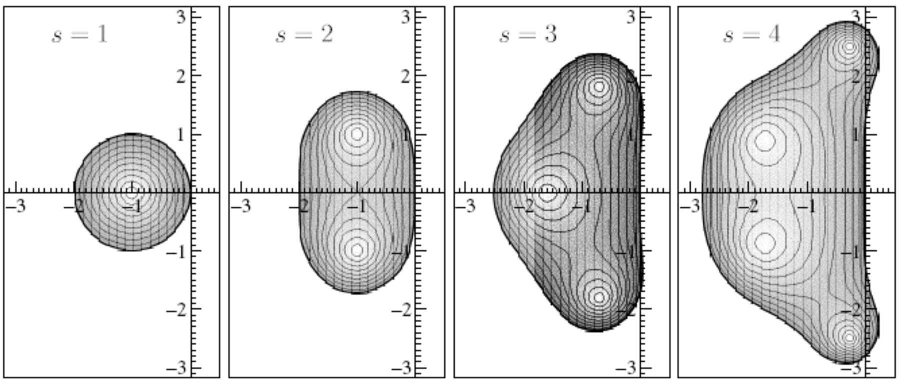
Problèmes raides

Stabilité : compléments

Pour une méthode ERK à s étapes

- $R(z)$ est un polynôme

  $\Rightarrow$ aucune méthode ERK ne peut être A-stable
- pour $s \leq 4$: $R(z) = 1 + z + \cdots + \frac{1}{s!}z^s$



Source : `http://www.unige.ch/~hairer/poly/chap3.pdf`

A-stabilité des méthodes IRK

A-stabilité of IRK Methods

Pour les méthodes IRK

- Euler implicite :

$$\mathscr{A} = \{z \in \mathbb{C} \text{ tels que } |z-1| > 1\}$$

- toutes les méthodes IRK ne sont pas A-stables
- toutes les méthodes de Runge-Kutta Gauss-Legendre sont A-stables

Problèmes à résoudre

Homework

1. Méthodes d'Adams-Bashforth et d'Adams-Moulton pour $q = 2$ (voir tableaux du premier cours).
   - Combien de pas ces méthodes comprennent-elles ?
   - Etablissez les polynômes caractéristiques pour chacune de ces deux méthodes. Déterminer leurs ordres respectifs à l'aide de ces polynômes.
2. Etablissez les expressions des polynômes ρ des méthodes d'Adams en général. Est-ce qu'elles sont toutes convergentes ?
3. La méthode saute-mouton est convergente, alors qu'elle n'est pas A-stable. Comment expliquez-vous cette contradiction (apparente) ?

Délai : 29 septembre 2014

Contrôle du pas de discrétisation
Discretisation Step Control

- Adaptation du pas de discrétisation k en fonction de l'erreur locale
- Deux types de stratégies (suivant le pas ou l'ordre)
  - comparer les erreurs respectives pour deux méthodes différentes de même ordre p et de constantes d'erreur connues
    $\rightarrow$ applicable avec méthodes multi-pas
  - avec une même méthode, comparer les erreurs pour deux pas différents (généralement k et $2k$)
    $\rightarrow$ applicable avec méthodes RK
  - comparer les erreurs respectives pour deux méthodes d'ordres p et q différents (généralement $q = p+1$)
    $\rightarrow$ applicable avec méthodes RK et multi-pas

Contrôle du pas de discrétisation (multi-pas)
Discretisation Step Control (Multi-step)

Méthodes numériques et éléments de programmation

Guy Munhoven

Rappels

Runge-Kutta

Stabilité absolue et A-Stabilité

Pas variables

Méth. multi-pas

Doublement

Ordres p, $p+1$

Problèmes raides

Stabilité : compléments

Soient deux méthodes multi-pas convergentes de même ordre p, de polynômes caractéristiques ρ, σ et $\tilde{\rho}, \tilde{\sigma}$ resp. :

$$\begin{cases} \rho(w) - \sigma(w)\ln(w) = c(w-1)^{p+1} + O(|w-1|)^{p+2}, \ w \to 1 \\ \tilde{\rho}(w) - \tilde{\sigma}(w)\ln(w) = \tilde{c}(w-1)^{p+1} + O(|w-1|)^{p+2}, \ w \to 1 \end{cases}$$

On peut montrer que l'erreur locale est telle que

$$\begin{cases} y(t_{j+1}) - y_{j+1} = c k^{p+1} y^{(p+1)}(t_{j+1}) + O(k^{p+2}) \\ y(t_{j+1}) - \tilde{y}_{j+1} = \tilde{c} k^{p+1} y^{(p+1)}(t_{j+1}) + O(k^{p+2}) \end{cases}$$

En soustrayant la deuxième de ces relations de la première, et en négligeant les termes $O(k^{p+2})$, nous pouvons estimer que

$$k^{p+1} y^{(p+1)}(t_{j+1}) \simeq \frac{\tilde{y}_{j+1} - y_{j+1}}{c - \tilde{c}}$$

Contrôle du pas de discrétisation (multi-pas)
Discretisation Step Control (Multi-step)

Il en résulte que

$$y(t_{j+1}) - y_{j+1} \simeq \frac{c}{c-\tilde{c}}(\tilde{y}_{j+1} - y_{j+1})$$

Estimateur pour l'erreur locale

$$\kappa = \left|\frac{c}{c-\tilde{c}}\right| |\tilde{y}_{j+1} - y_{j+1}|$$

Critère de contrôle du pas :

$$\kappa < k\delta$$

Motivation : Erreur par unité de pas doit rester en-dessous d'une certaine limite δ fixée préalablement.

Contrôle du pas de discrétisation (multi-pas)
Discretisation Step Control (Multi-step)

Méthodes numériques et éléments de programmation

Guy Munhoven

Rappels
Runge-Kutta
Stabilité absolue et A-Stabilité
Pas variables
Méth. multi-pas
Doublement
Ordres p, $p+1$
Problèmes raides
Stabilité : compléments

Réalisation pratique

- $\kappa > k\delta$: rejeter y_{j+1} calculé, diminuer le pas de moitié, calculer des nouvelles valeurs initiales par interpolation et répéter
- $k\delta/10 < \kappa \leq k\delta$: continuer sans modification
- $\kappa \leq k\delta/10$: accepter y_{j+1} calculé, doubler le pas, ajuster les valeurs initiales (en supprimer une sur deux et compléter par des plus anciennes)

Inconvénient : exige de garder en mémoire au moins $2s-1$ valeurs précédentes

Contrôle du pas de discrétisation : doublement

Discretisation Step Control : Step Doubling

- Progression d'un méthode de Runge-Kutta

$$y_{j+1} = y_j + k\Phi(t_j, y_j, k)$$

- Erreur de discrétisation locale d'une méthode d'ordre p

$$y(t_{j+1}) - y_{j+1} = c(y(t_j))k^{p+1} + O(k^{p+2})$$

- Deux pas successifs de t_{j-1} à t_j et t_{j+1} mènent de $y(t_{j-1})$ (exact) à $y_j \simeq y(t_j)$ et $y_{j+1} \simeq y(t_{j+1})$ (approximations) ; un seul pas de t_j à t_{j+1} mène de $y(t_j)$ à $y^*_{j+1} \simeq y(t_{j+1})$

$$\begin{array}{lcllll} y_j & = & y(t_{j-1}) & + & k\Phi(t_{j-1}, y(t_{j-1}), k) \\ y_{j+1} & = & y_j & + & k\Phi(t_j, y_j, k) \\ y^*_{j+1} & = & y(t_j) & + & k\Phi(t_j, y(t_j), k) \end{array}$$

Méthodes numériques et éléments de programmation

Guy Munhoven

Rappels

Runge-Kutta

Stabilité absolue et A-Stabilité

Pas variables

Méth. multi-pas

Doublement

Ordres p, $p+1$

Problèmes raides

Stabilité : compléments

Contrôle du pas de discrétisation : doublement

Discretisation Step Control : Step Doubling

Problème : évaluer le terme principal de $y(t_{j+1}) - y_{j+1}$
Il vient

$$\begin{aligned} & y(t_{j+1}) - y_{j+1} \\ & \quad = \quad y(t_{j+1}) - y^*_{j+1} + y^*_{j+1} - y_{j+1} \\ & \quad = \quad c(y(t_j))k^{p+1} + O(k^{p+2}) \\ & \qquad\quad + y(t_j) + k\Phi(t_j, y(t_j), k) - y_j - k\Phi(t_j, y_j, k) \\ & \quad = \quad \left(c(y(t_{j-1})) + k \left.\frac{dc}{dy}\right|_{y(t_{j-1})} y'(t_{j-1}) + O(k^2)\right)k^{p+1} \\ & \qquad\quad + c(y(t_{j-1}))k^{p+1} + O(k^{p+2}) + kO(y(t_j) - y_j) \\ & \quad = \quad 2c(y(t_{j-1}))k^{p+1} + O(k^{p+2}) + kO(k^{p+1}) \end{aligned}$$

Contrôle du pas de discrétisation : doublement

Discretisation Step Control : Step Doubling

Ainsi, pour deux pas successifs

$$y(t_{j+1}) - y_{j+1} = 2c(y(t_{j-1}))k^{p+1} + O(k^{p+2}) \qquad (*)$$

Pour un pas de longueur $2k$ de t_{j-1} vers t_{j+1}, menant de $y_{j-1} = y(t_{j-1})$ vers $\tilde{y}_{j+1} \simeq y(t_{j+1})$:

$$y(t_{j+1}) - \tilde{y}_{j+1} = c(y(t_{j-1}))(2k)^{p+1} + O(k^{p+2}),$$

que nous pouvons réorganiser comme suit

$$y(t_{j+1}) - \tilde{y}_{j+1} = 2^p \times 2c(y(t_{j-1}))k^{p+1} + O(k^{p+2}). \qquad (**)$$

En retranchant $(**)$ de $(*)$, nous trouvons que

$$\kappa = |2c(y(t_{j-1}))k^{p+1}| = \simeq \frac{|\tilde{y}_{j+1} - y_{j+1}|}{2^p - 1}$$

Méthodes numériques et éléments de programmation

Guy Munhoven

Rappels
Runge-Kutta
Stabilité absolue et A-Stabilité
Pas variables
Méth. multi-pas
Doublement
Ordres p, $p+1$
Problèmes raides
Stabilité : compléments

Contrôle du pas de discrétisation : adaptation

Discretisation Step Control : Step Adaptation

- Critère d'ajustement : comparer $\kappa = \kappa(k) = \frac{|\tilde{y}_{j+1} - y_{j+1}|}{2^p - 1}$ à $k\delta$, où δ est une tolérance par unité de pas pré-définie
  - si $\kappa(k) > k\delta$: rejeter y_{j+1} et re-calculer une nouvelle estimation avec un pas réduit
  - si $\kappa(k) < k\delta$: accepter y_{j+1} et augmenter le pas pour l'itération suivante
- Pour deux longueurs de pas différentes, k et k_c, les erreurs respectives affectant y_{j+1} sont telles que

$$\frac{\kappa(k_c)}{\kappa(k)} \simeq \left(\frac{k_c}{k}\right)^{p+1}$$

- Pour avoir $\kappa(k_c) < k_c\delta$ il suffirait donc que

$$\kappa(k_c) \simeq \kappa(k)\left(\frac{k_c}{k}\right)^{p+1} < k_c\delta$$

  c'est-à-dire que

$$k_c < k\left(\frac{k\delta}{\kappa(k)}\right)^{\frac{1}{p}}$$

- En pratique, un facteur de sécurité de 0.8 est appliqué à cette borne et on demande que $k/2 < k_c < 2k$

- Procédure d'ajustement permet un contrôle du pas très fin avec les méthodes de Runge-Kutta.
- Désavantage : la validation d'un pas demande presque autant de d'opérations que trois pas directs.
- Critère d'ajustement : quelques remarques
  - on peut aussi adopter une *tolérance absolue* au lieu d'une *tolérance par unité de pas*, et donc comparer κ à δ au lieu de $k\delta$, et de baser l'ajustement du pas sur $\kappa(k_c) < \delta$;
  - la facteur de sécurité 0.8 est parfois porté à 0.9 ;
  - la gamme de pas acceptés au cours d'une modification est parfois étendue à l'intervalle $]k/5,\ 5k[$.

Contrôle du pas de discrétisation : ordre
Discretisation Step Control : Order

- Pour deux méthodes d'ordres p et $\tilde{p} = p+1$, les erreurs de discrétisation locales sont

$$\begin{cases} y(t_{j+1}) - y_{j+1} & = & c(y(t_j))k^{p+1} + O(k^{p+2}) \\ y(t_{j+1}) - \tilde{y}_{j+1} & = & O(k^{p+2}) \end{cases}$$

- L'erreur qui affecte y_{j+1} est donc

$$\kappa = |c(y(t_j))k^{p+1}| = \simeq |\tilde{y}_{j+1} - y_{j+1}|$$

- Ajustement de k_c comme pour ≪Doublement de pas≫
- Méthode de choix pour les méthodes de Runge-Kutta *emboîtées* : avec un même tableau (a_{mn}) on dérive deux méthodes différentes en utilisant des b_m différents

Méthodes de Runge-Kutta emboîtées
Embedding Runge-Kutta Methods

RK-Fehlberg 2(3) — 3 étapes

$$\begin{array}{c|ccc} 0 & \cdot & \cdot & \cdot \\ 1 & 1 & \cdot & \cdot \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & \cdot \\ \hline p=2 & \frac{1}{2} & \frac{1}{2} & 0 \\ p=3 & \frac{1}{6} & \frac{1}{6} & \frac{2}{3} \end{array}$$

Méthodes de Runge-Kutta emboîtées

Embedding Runge-Kutta Methods

RK-Fehlberg 4(5) — 6 étapes

$$
\begin{array}{c|cccccc}
0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\frac{1}{4} & \frac{1}{4} & \cdot & \cdot & \cdot & \cdot & \cdot \\
\frac{3}{8} & \frac{3}{32} & \frac{9}{32} & \cdot & \cdot & \cdot & \cdot \\
\frac{12}{13} & \frac{1932}{2197} & \frac{7296}{2197} & 0 & \cdot & \cdot & \cdot \\
1 & \frac{439}{216} & -8 & \frac{3680}{513} & -\frac{845}{4104} & \cdot & \cdot \\
\frac{1}{2} & -\frac{8}{27} & 2 & \frac{3544}{2565} & -\frac{1859}{4104} & -\frac{11}{40} & \cdot \\
\hline
p=4 & \frac{25}{216} & 0 & \frac{1408}{2565} & \frac{2197}{4104} & -\frac{1}{5} & 0 \\
p=5 & \frac{16}{135} & 0 & \frac{6656}{12855} & \frac{28561}{56430} & -\frac{9}{50} & \frac{2}{55}
\end{array}
$$

A noter que les ordres des deux méthodes sont inférieures à 6, le nombre d'étapes.

Méthodes de Runge-Kutta emboîtées

Embedding Runge-Kutta Methods

Dormand-Prince 5(4) — 7 étapes

$$
\begin{array}{c|ccccccc}
0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\frac{1}{5} & \frac{1}{5} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\frac{3}{10} & \frac{3}{40} & \frac{9}{40} & \cdot & \cdot & \cdot & \cdot & \cdot \\
\frac{4}{5} & \frac{44}{55} & -\frac{56}{15} & 0 & \cdot & \cdot & \cdot & \cdot \\
\frac{8}{9} & \frac{19372}{6561} & -\frac{25360}{2187} & \frac{64448}{6561} & -\frac{212}{729} & \cdot & \cdot & \cdot \\
1 & \frac{9017}{3168} & -\frac{355}{33} & \frac{46732}{5247} & \frac{49}{176} & -\frac{5103}{18656} & \cdot & \cdot \\
1 & \frac{35}{384} & 0 & \frac{500}{1113} & \frac{125}{192} & -\frac{2187}{6784} & \frac{11}{84} & \cdot \\
\hline
p=4 & \frac{5179}{57600} & 0 & \frac{7571}{16695} & \frac{393}{640} & -\frac{92097}{339200} & \frac{187}{2100} & \frac{1}{40} \\
p=5 & \frac{35}{384} & 0 & \frac{500}{1113} & \frac{125}{192} & -\frac{2187}{6784} & \frac{11}{84} & 0
\end{array}
$$

Méthodes de Runge-Kutta emboîtées

Embedding Runge-Kutta Methods

- Autres : RK–Dormand-Prince 8(6), RK–Verner 5(4), ...
- Notation des tableaux de Butcher étendus pour les méthodes de Runge-Kutta emboîtées pas standardisée.
- Généralement : la première ligne de coefficients b_m correspond à la méthode d'ordre inférieur.
- La dénomination de la méthode précise les rôles de chaque méthode individuelle : pour *Méthode $p_1(p_2)$*,
  - la méthode d'ordre p_1 fournit l'approximation à adopter en fin de pas,
  - la méthode d'ordre p_2 n'est utilisée que pour le monitoring de l'erreur,
  - le plus petit des p_1, p_2 est utilisé à la place de p pour l'ajustement du pas.

Problèmes raides (stiff)

Stiff Problems

Exemple :

$$\begin{aligned} y_1' &= -101y_1 \quad -99y_2, \qquad y_1(0)=2 \\ y_2' &= -99y_1 \quad -101y_2, \qquad y_2(0)=0 \end{aligned}$$

Solution analytique :

$$\begin{aligned} y_1(t) &= e^{-200t} + e^{-2t} \\ y_2(t) &= e^{-200t} - e^{-2t} \end{aligned}$$

Problèmes raides (stiff)

Stiff Problems

- Euler explicite :

$$\begin{aligned} y_{1,j+1} &= y_{1,j} + k(-101 y_{1,j} - 99 y_{2,j}) \\ y_{2,j+1} &= y_{2,j} + k(-99 y_{1,j} - 101 y_{2,j}) \end{aligned}$$

- Dans cet exemple particulier, nous avons

$$\begin{aligned} y_{1,j} &= (1-200k)^j + (1-2k)^j \\ y_{2,j} &= (1-200k)^j - (1-2k)^j \end{aligned}$$

- Pour des raisons de stabilité, il faut que $|1-200k| < 1$ et $|1-2k| < 1$ *en permanence*, c'est-à-dire, que $k < 0.01$, alors que pour $t > 0.05$, $e^{-200t}/e^{-2t} < \frac{1}{10000}$ rendant e^{-200t} négligeable.

Problèmes raides (stiff)

Stiff Problems

- Euler implicite :

$$\begin{aligned} y_{1,j+1} &= y_{1,j} + k(-101 y_{1,j+1} - 99 y_{2,j+1}) \\ y_{2,j+1} &= y_{2,j} + k(-99 y_{1,j+1} - 101 y_{2,j+1}) \end{aligned}$$

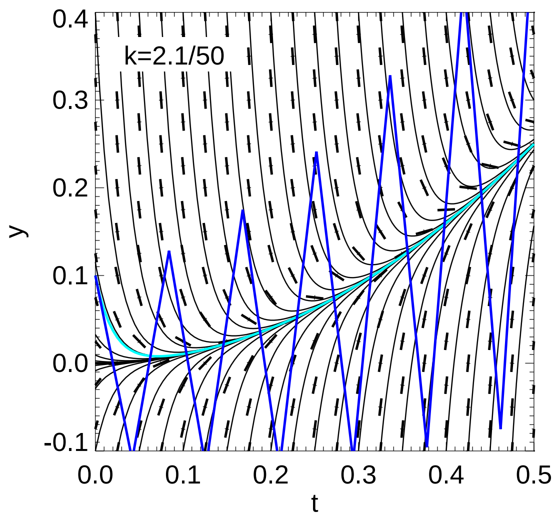
- Dans cet exemple particulier, nous avons

$$\begin{aligned} y_{1,j} &= (1+200k)^{-j} + (1+2k)^{-j} \\ y_{2,j} &= (1+200k)^{-j} - (1+2k)^{-j} \end{aligned}$$

- Stabilité garantie pour tout $k > 0$

$$y' = 50(t^2 - y) + 2t, \quad y(0) = 0.1$$

Euler explicite



Problèmes raides (stiff)
Stiff Problems

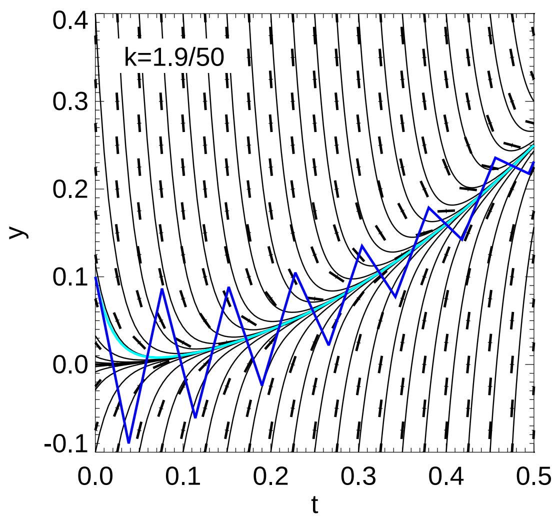
Méthodes numériques et éléments de programmation

Guy Munhoven

Rappels
Runge-Kutta
Stabilité absolue et A-Stabilité
Pas variables
Méth. multi-pas
Doublement
Ordres p, $p+1$
Problèmes raides
Stabilité : compléments

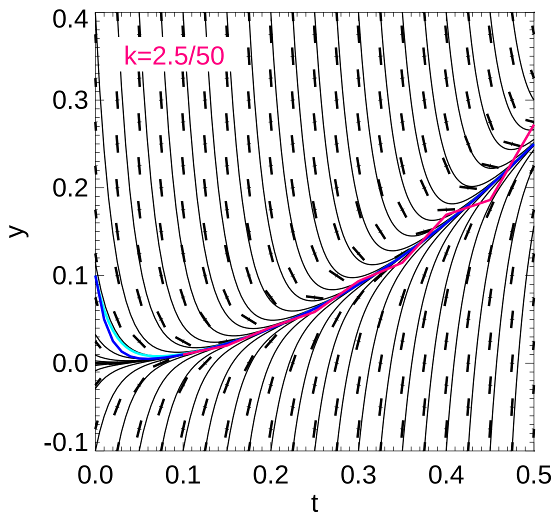
$$y' = 50(t^2 - y) + 2t, \quad y(0) = 0.1$$

Euler explicite



$$y' = 50(t^2 - y) + 2t, \quad y(0) = 0.1$$

Euler explicite



Problèmes raides (stiff)
Stiff Problems

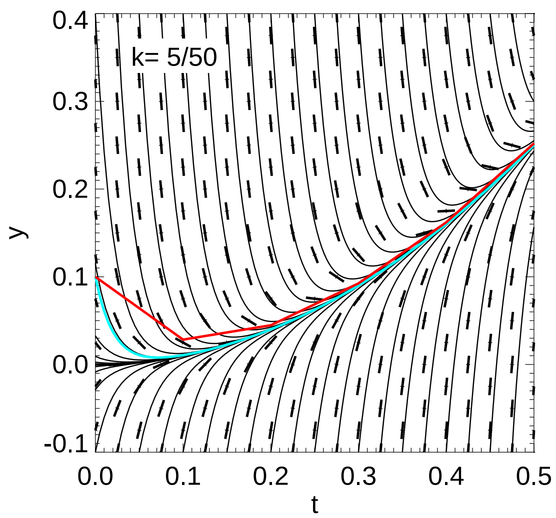
Méthodes numériques et éléments de programmation

Guy Munhoven

Rappels
Runge-Kutta
Stabilité absolue et A-Stabilité
Pas variables
Méth. multi-pas
Doublement
Ordres p, $p+1$
Problèmes raides
Stabilité : compléments

$$y' = 50(t^2 - y) + 2t, \quad y(0) = 0.1$$

Euler implicite

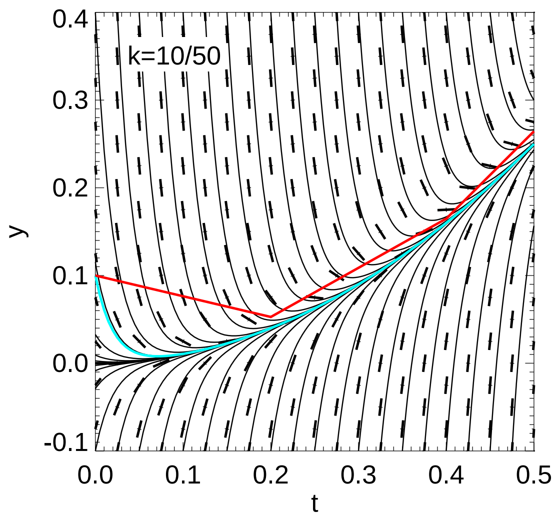


Problèmes raides (stiff)

Stiff Problems

$$y' = 50(t^2 - y) + 2t, \quad y(0) = 0.1$$

Euler implicite



Problèmes raides (stiff) : méthodes BDF

Stiff Problems : BDF Methods

- Méthodes implicites avantageuses
- Formules de différentiation rétrograde (*BDF* – Backward Differentiation Formulae)
  - Méthodes multi-pas
  - Basées sur des approximations de la dérivée de y au noeud t_{j+1}
  - Considérer le polynôme P de degré $q > 0$ passant par (t_{j+1}, y_{j+1}) et les q points $(t_j, y_j), \ldots, (t_{j+1-q}, y_{j+1-q})$ qui le précèdent et telque $P'(t_{j+1}) = y'_{j+1} = f_{j+1}$
  - Définition implicite

Problèmes raides (stiff) : méthodes BDF
Stiff Problems : BDF Methods

Méthodes BDF

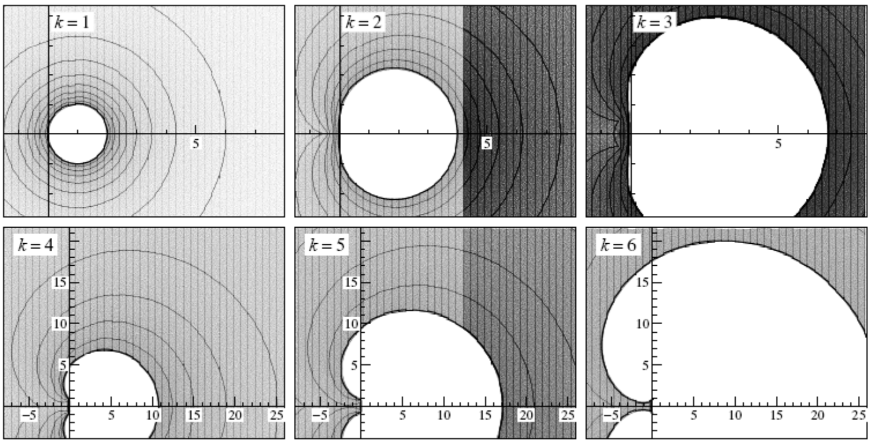
q	Itération pour calculer y_{j+1}
1	$y_{j+1} - y_j = kf_{j+1}$ *(Euler implicite)*
2	$y_{j+1} - \frac{4}{3}y_j + \frac{1}{3}y_{j-1} = \frac{2}{3}kf_{j+1}$
3	$y_{j+1} - \frac{18}{11}y_j + \frac{9}{11}y_{j-1} - \frac{2}{11}y_{j-2} = \frac{6}{11}kf_{j+1}$
4	$y_{j+1} - \frac{48}{25}y_j + \frac{36}{25}y_{j-1} - \frac{16}{25}y_{j-2} + \frac{3}{25}y_{j-3} = \frac{12}{25}kf_{j+1}$
5	$y_{j+1} - \frac{300}{137}y_j + \frac{300}{137}y_{j-1} - \frac{200}{137}y_{j-2} + \frac{75}{137}y_{j-3} - \frac{12}{137}y_{j-4} = \frac{60}{137}kf_{j+1}$
6	$y_{j+1} - \frac{360}{147}y_j + \frac{450}{147}y_{j-1} - \frac{400}{147}y_{j-2} + \frac{225}{147}y_{j-3} - \frac{72}{147}y_{j-4} + \frac{10}{147}y_{j-5} = \frac{60}{147}kf_{j+1}$

Erreur de troncature (locale) : $O(k^q)$

Problèmes raides (stiff) : méthodes BDF
Stiff Problems : BDF Methods

- $\sigma(w) = \beta w^q$, $1/\beta = \sum_{m=1}^{q} \frac{1}{m}$,
  $\rho(w) = \beta \sum_{m=1}^{q} \frac{1}{m} w^{q-m}(w-1)^m$
- A-stables et convergentes pour $q = 1$ et 2
- A(α)-stables et convergentes pour $q = 3, \ldots, 6$
- non convergentes pour $q > 6$
- `http://scholarpedia.org/article/Backward_Differentiation_Formulas`

Problèmes raides (stiff) : méthodes BDF
Stiff Problems : BDF Methods



Source : `http://www.unige.ch/~hairer/poly/chap3.pdf`

Stabilité : quelques compléments
Stability : a few complements

Méthodes numériques et éléments de programmation
Guy Munhoven
Rappels
Runge-Kutta
Stabilité absolue et A-Stabilité
Pas variables
Méth. multi-pas
Doublement
Ordres p, $p+1$
Problèmes raides
Stabilité : compléments

- Stabilité absolue ou A-stabilité d'une méthode
  - pour l'équation linéaire $y' = \lambda y$
  - pour k fixé
  - y_j doit rester borné quand $t_j \to \infty$
- Stabilité ou zéro-stabilité d'une méthode
  - pour une E.D.O. bien posée quelconque
  - pour un intervalle d'intégration fixé (de longueur T)
  - définit le comportement de la méthode soumise à des perturbations suffisamment petites (p.ex., résultant d'erreurs d'arrondi)
  - il doit exister $k_0 > 0$ et $C > 0$ tels que les approximations z_j^k ($j = 0, \ldots, T/k$) solutions de la méthode soumise à des perturbations δ_j suffisamment petites (si $|\delta_j| < \varepsilon, \forall \varepsilon \ll$) doivent pouvoir être ramenées arbitrairement proches des solutions y_j^k ($j = 0, \ldots, T/k$) de la méthode non perturbée ($|z_j^k - y_j^k| < C\varepsilon, j = 0, \ldots, T/k$), dès que $k < k_0$.

Stabilité : quelques compléments

Stability : a few complements

Pour une méthode *consistante*

- zéro-stabilité $\Leftrightarrow$ convergence
- stabilité absolue $\Rightarrow$ zéro-stabilité
- il existe des méthodes zéro-stables non A-stables

Dans le cas d'une méthode multi-pas consistante

- condition de racine $\Leftrightarrow$ zéro-stabilité