

Intégration en temps de systèmes raides par des méthodes à un pas

Laurent Séries Marc Massot

CMAP, Ecole Polytechnique

MAP 551 - Cours 5

Plan de la présentation

- 1 Rappel sur les limitations des méthodes classiques
 - La notion de A-stabilité
 - Les méthodes de Runge-Kutta implicite
- 2 La méthode RadauIIA
 - La méthode RadauIIA
 - Reformulation du système et méthode de Newton simplifiée
 - Résolution du système linéaire
 - Contrôle du pas
- 3 Résolution de l'équation de la chaleur
 - Méthodes implicites
 - Implémentation
 - Méthodes explicites

Notion de raideur

Raideur : problème rencontré par les méthodes numériques explicites lors de l'intégration numérique du système.

Elle est liée à deux aspects :

- Le spectre de la matrice jacobienne associée au système (notamment, la dispersion des valeurs propres)
- La "distance" de la condition initiale à la "variété d'équilibre" du système

Illustration sur un exemple : Curtiss & Hirschfelder, 1952

Notion de raideur

Raideur : problème rencontré par les méthodes numériques explicites lors de l'intégration numérique du système. Exemple : Curtiss & Hirschfelder, 1952

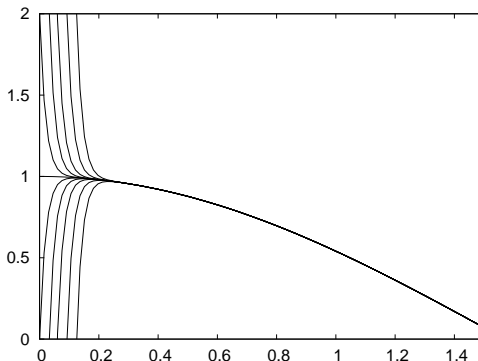
$$y' = \frac{-y + \cos t}{\varepsilon} \quad 0 < \varepsilon \ll 1$$

Solutions :

$$y(t) = Ce^{-t/\varepsilon} + \frac{\cos t}{1 + \varepsilon^2} + \frac{\varepsilon \cos t}{1 + \varepsilon^2} \quad C \in \mathbb{R}$$

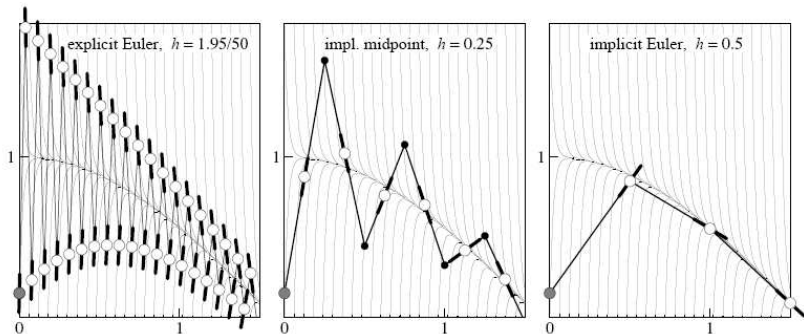
Notion de raideur

Raideur : problème rencontré par les méthodes numériques explicites lors de l'intégration numérique du système. Exemple : Curtiss & Hirschfelder, 1952
Solutions ($\varepsilon = 1/50$) :



Notion de raideur

Raideur : problème rencontré par les méthodes numériques explicites lors de l'intégration numérique du système. Différentes méthodes numériques



Cadre général d'une équation différentielle ordinaire

On cherche à résoudre une équation différentielle ordinaire de la forme

$$y' = f(t, y), \quad y(t_0) = y_0,$$

y étant un vecteur de \mathbb{R}^n et f une fonction de $\mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$. On suppose avoir toute la régularité possible sur la solution.

- On découpe l'intervalle sur lequel on veut résoudre l'équation différentielle,
- On cherche ensuite comment calculer en un point de cet intervalle une valeur approchée de la solution.

Cadre général d'une équation différentielle ordinaire

On cherche à résoudre une équation différentielle ordinaire de la forme

$$y' = f(t, y), \quad y(t_0) = y_0,$$

y étant un vecteur de \mathbb{R}^n et f une fonction de $\mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$. On suppose avoir toute la régularité possible sur la solution.

- On découpe l'intervalle sur lequel on veut résoudre l'équation différentielle,
- On cherche ensuite comment calculer en un point de cet intervalle une valeur approchée de la solution.

Cadre général d'une équation différentielle ordinaire

On cherche à résoudre une équation différentielle ordinaire de la forme

$$y' = f(t, y), \quad y(t_0) = y_0,$$

y étant un vecteur de \mathbb{R}^n et f une fonction de $\mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$. On suppose avoir toute la régularité possible sur la solution.

- On découpe l'intervalle sur lequel on veut résoudre l'équation différentielle,
- On cherche ensuite comment calculer en un point de cet intervalle une valeur approchée de la solution.

Au fil de l'histoire

- Méthode d'Euler (1768)

$$y_{n+1} = y_n + h_n f(t_n, y_n)$$

- Pour obtenir une meilleure approximation, il faut se demander comment approcher une intégrale de manière précise....
- Méthode de Runge (1895)

$$y_{n+1} = y_n + h_n f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2} f(t_n, y_n)\right)$$

Au fil de l'histoire

- Méthode d'Euler (1768)

$$y_{n+1} = y_n + h_n f(t_n, y_n)$$

- Pour obtenir une meilleure approximation, il faut se demander comment approcher une intégrale de manière précise....
- Méthode de Runge (1895)

$$y_{n+1} = y_n + h_n f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2} f(t_n, y_n)\right)$$

Au fil de l'histoire

- Méthode d'Euler (1768)

$$y_{n+1} = y_n + h_n f(t_n, y_n)$$

- Pour obtenir une meilleure approximation, il faut se demander comment approcher une intégrale de manière précise....
- Méthode de Runge (1895)

$$y_{n+1} = y_n + h_n f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2} f(t_n, y_n)\right)$$

Au fil de l'histoire

- Méthode de Heun (1900), on pose

$$k = f\left(t_n + \frac{h}{3}, y_n + \frac{h}{3}f(t_n, y_n)\right),$$

alors

$$y_{n+1} = y_n + \frac{h_n}{4} \left(f(t_n, y_n) + 3f\left(t_n + \frac{2h}{3}, y_n + \frac{2h}{3}k\right) \right).$$

- Procédé qui se généralise, les méthodes de Runge-Kutta sont nées...

Au fil de l'histoire

- Méthode de Heun (1900), on pose

$$k = f\left(t_n + \frac{h}{3}, y_n + \frac{h}{3}f(t_n, y_n)\right),$$

alors

$$y_{n+1} = y_n + \frac{h_n}{4} \left(f(t_n, y_n) + 3f\left(t_n + \frac{2h}{3}, y_n + \frac{2h}{3}k\right) \right).$$

- Procédé qui se généralise, les méthodes de Runge-Kutta sont nées...

Au fil de l'histoire

Kutta (1901) :

Une méthodes de Runge-Kutta à s étages est donnée par
(on simplifie en écrivant la première itération)

$$k_1 = f(t_0, y_0),$$

$$k_2 = f(t_0 + c_2 h, y_0 + h a_{21} k_1),$$

$$k_3 = f(t_0 + c_3 h, y_0 + h a_{31} k_1 + h a_{32} k_2),$$

...

$$k_s = f(t_0 + c_s h, y_0 + h a_{s1} k_1 + h a_{s,s-1} k_{s-1}),$$

$$y_1 = y_0 + h(b_1 k_1 + \dots b_s k_s).$$

Ecriture sous forme d'un tableau !

Au fil de l'histoire

Méthode de Heun

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

"La" méthode de Runge-Kutta

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

Au fil de l'histoire

Méthode de Heun

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

"La" méthode de Runge-Kutta

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

Définition

On suppose le pas constant égal à h .

On dit qu'une méthode de Runge-Kutta est d'ordre p si, pour chaque problème $y' = f(t, y)$, $y(t_0) = y_0$, l'erreur, après un pas, satisfait pour h petit tendant vers 0,

$$y_1 - y(t_0 + h) = \mathcal{O}(h^{p+1})$$

Le terme en rouge s'appelle l'erreur locale de troncature.

Exemple : La méthode d'Euler est d'ordre 1, "la" méthode de Runge-Kutta est d'ordre 4. On aime pour un ordre fixé, choisir la méthode qui fait le moins d'évaluations...

Définition

On suppose le pas constant égal à h .

On dit qu'une méthode de Runge-Kutta est d'ordre p si, pour chaque problème $y' = f(t, y)$, $y(t_0) = y_0$, l'erreur, après un pas, satisfait pour h petit tendant vers 0,

$$y_1 - y(t_0 + h) = \mathcal{O}(h^{p+1})$$

Le terme en rouge s'appelle l'erreur locale de troncature.

Exemple : La méthode d'Euler est d'ordre 1, "la" méthode de Runge-Kutta est d'ordre 4. On aime pour un ordre fixé, choisir la méthode qui fait le moins d'évaluations...

Résultat

Si f satisfait une condition de Lipschitz uniformément par rapport à la deuxième variable dans un voisinage de la solution de $y' = f(t, y)$ alors on a une estimation de la forme

$$\|y(t_n) - y_n\| \leq Ch^p.$$

Pas de temps variable

Un calcul à pas constants est en général inefficace. L'idée est de choisir les pas afin que l'erreur locale soit partout environ égale à une tolérance *Tol* donnée.

On utilise pour cela **deux** méthodes de Runge-Kutta d'ordre p et \hat{p} avec $\hat{p} < p$, donnant comme approximations y_1 et \hat{y}_1 . Le h optimal s'obtient à partir de h avec la formule

$$h_{\text{opt}} = 0,9.h. \sqrt[p+1]{\frac{\text{Tol}}{\|y_1 - \hat{y}_1\|}}$$

Pour limiter les calculs, on utilise des méthodes **emboîtées** (exemple : Euler et Runge).

Fonction de stabilité

Soit λ un nombre complexe, on considère le problème test de Dahlquist, l'équation différentielle complexe :

$$\dot{u} = \lambda u. \quad (1)$$

Sa solution exacte est

$$y(t) = e^{\lambda t} C$$

et elle reste bornée pour $t \geq 0$ si $\Re \lambda \leq 0$.

Il s'agit d'une équation différentielle très simple mais qui peut donner des informations importantes sur le comportement des schémas numériques.

La solution numérique d'une méthode de Runge Kutta appliquée avec des pas constants ne dépend que du produit $h\lambda$. On écrit les itérations sous la forme

$$y_{n+1} = R(h\lambda)y_n. \quad (2)$$

R est appelée la fonction de stabilité de la méthode, il s'agit d'un polynôme.

Fonction de stabilité

On note

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

Cet ensemble est appelé domaine de stabilité de la méthode.

On dit que la méthode est A-stable si

$$\{z \in \mathbb{C} : \Re z \leq 0\} \subset \mathcal{S}.$$

Une méthode de Runge-Kutta explicite n'est jamais A-stable.

Fonction de stabilité

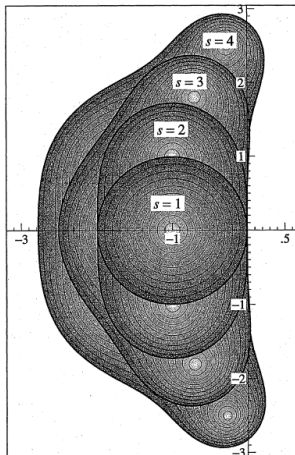


Fig. 2.1. Stability domains for explicit Runge-Kutta methods of order $p = s$

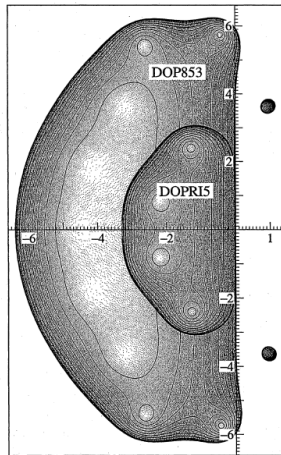


Fig. 2.2. Stability domains for DOPRI methods

Plan de la présentation

- 1 Rappel sur les limitations des méthodes classiques
 - La notion de A-stabilité
 - Les méthodes de Runge-Kutta implicite
- 2 La méthode RadauIIA
 - La méthode RadauIIA
 - Reformulation du système et méthode de Newton simplifiée
 - Résolution du système linéaire
 - Contrôle du pas
- 3 Résolution de l'équation de la chaleur
 - Méthodes implicites
 - Implémentation
 - Méthodes explicites

Domaine de stabilité - A-stabilité

Definition (A-stabilité)

Considérons une méthode dont la solution numérique $y_{n \geq 0}$ pour l'équation de test est une fonction de $z = h\lambda$. Alors l'ensemble

$$S := \{z \in \mathbb{C}; \{y_n\}_{n \geq 0} \text{ est bornée}\} \quad (3)$$

s'appelle domaine de stabilité de la méthode. On dit que la méthode est A-stable si

$$S \supset \mathbb{C}^- \text{ où } \mathbb{C}^- = \{z \in \mathbb{C}; \Re z \leq 0\} \quad (4)$$

Classes de méthodes

- Dans le cas des méthodes à un pas explicites, le domaine de stabilité S est borné et la condition de stabilité $h\lambda \in S$ impose une **restriction sévère à h** . Ces méthodes ne sont donc pas appropriées pour les problèmes raides.
- Les méthodes d'Adams explicites et implicites (à l'exception de la méthode d'Euler implicite et de la règle du trapèze) ont toutes un domaine de stabilité borné et petit. Ces méthodes ne sont donc **pas utilisables pour des problèmes raides**.
- Les méthodes BDF, par contre, ont un domaine de stabilité plus important, ce qui explique qu'elles sont beaucoup utilisées pour résoudre des problèmes raides.
La méthode BDF avec $k = 2$ est même A-stable.
- La **barrière de Dahlquist** dit que l'ordre d'une méthode multipas A-stable ne peut être plus grand que 2. Ce résultat a contribué à la détermination d'autres méthodes d'intégration permettant de combiner A-stabilité et ordre élevé.

→ Besoin de créer des méthodes à un pas très stable et d'ordre élevé :

Runge Kutta implicite

Plan de la présentation

- 1 **Rappel sur les limitations des méthodes classiques**
 - La notion de A-stabilité
 - Les méthodes de Runge-Kutta implicite
- 2 La méthode RadauIIA
 - La méthode RadauIIA
 - Reformulation du système et méthode de Newton simplifiée
 - Résolution du système linéaire
 - Contrôle du pas
- 3 Résolution de l'équation de la chaleur
 - Méthodes implicites
 - Implémentation
 - Méthodes explicites

IRK

Definition (Méthode de Runge-Kutta à s étapes)

Soient b_i, a_{ij} ($i, j = 1 \dots, s$) des réels. La méthode de Runge-Kutta à s étapes s'écrit :

$$k_i = f(x_0 + c_i h, y_0 + h \sum_{j=1}^s a_{ij} k_j) \quad i = 1, \dots, s$$

$$y_1 = y_0 + h \sum_{i=1}^s b_i k_i$$

Quand $a_{ij} = 0$ pour $i \geq j$, la méthode est explicite. Si $a_{ij} = 0$ pour $i > j$, et qu'au moins un des $a_{ij} \neq 0$, la méthode est dite diagonalement implicite (diagonal implicit RK, DIRK). Si de plus tous les termes diagonaux sont égaux $a_{ii} = \gamma$ pour $i = 1, \dots, s$, on parle de "singly diagonal implicit method" (SDIRK). Dans tous les autres cas, la méthode est implicite.

Stabilité et ordre des IRK

$$\begin{aligned} B(p) : \quad \sum_{i=1}^s b_i c_i^{q-1} &= \frac{1}{q} & q = 1, \dots, p \\ C(\eta) : \quad \sum_{j=1}^s a_{ij} c_j^{q-1} &= \frac{c_i^q}{q} & i = 1, \dots, s, \quad q = 1, \dots, \eta \\ D(\zeta) : \quad \sum_{i=1}^s b_i c_i^{q-1} a_{ij} &= \frac{b_j}{q} (1 - c_j^q) & j = 1, \dots, s, \quad q = 1, \dots, \zeta \end{aligned}$$

La condition $B(p)$ signifie que la formule de quadrature (b_i, c_i) est d'ordre p .

Théorème (Butcher)

Si les coefficients b_i , c_i , et a_{ij} de la méthode de RK vérifie les hypothèses $B(p)$, $C(\eta)$, et $D(\zeta)$ avec $p \leq \eta + \zeta + 1$ et $p \leq 2\eta + 2$ alors la méthode est d'ordre p .

Plan de la présentation

- 1 Rappel sur les limitations des méthodes classiques
 - La notion de A-stabilité
 - Les méthodes de Runge-Kutta implicite
- 2 **La méthode RadauIIA**
 - **La méthode RadauIIA**
 - Reformulation du système et méthode de Newton simplifiée
 - Résolution du système linéaire
 - Contrôle du pas
- 3 Résolution de l'équation de la chaleur
 - Méthodes implicites
 - Implémentation
 - Méthodes explicites

Origine de la méthode

- Ces méthodes ne sont pas forcément A-stables. Ehle reprend les idées de Butcher et construit des méthodes de type I, II et III avec d'excellentes conditions de stabilité.
- La méthode d'Ehle de type II, appelée méthode de Radau IIA, s'obtient par application de la condition $C(s)$. On peut montrer qu'elle constitue ainsi une méthode de collocation basé sur les zéros de la fonction $\frac{d^{s-1}}{dx^{s-1}}(x^{s-1}(x-1)^s)$. Elle est d'ordre $2s - 1$ et est A-stable
- cette méthode pour les valeurs $s = 3$ et $p = 5$ est aussi L-stable et est implémentée dans le code radau5.

Coefficients de RadauIIA

- cette méthode pour les valeurs $s = 3$ et $p = 5$ est aussi L-stable et est implémentée dans le code Radau5.

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

Plan de la présentation

- 1 Rappel sur les limitations des méthodes classiques
 - La notion de A-stabilité
 - Les méthodes de Runge-Kutta implicite
- 2 **La méthode RadauIIA**
 - La méthode RadauIIA
 - **Reformulation du système et méthode de Newton simplifiée**
 - Résolution du système linéaire
 - Contrôle du pas
- 3 Résolution de l'équation de la chaleur
 - Méthodes implicites
 - Implémentation
 - Méthodes explicites

Réécriture de la méthode

$$\begin{aligned}g_i &= y_0 + h \sum_{j=1}^s a_{ij} f(x_0 + c_j h, g_j) \quad i = 1, \dots, s \\ y_1 &= y_0 + h \sum_{j=1}^s b_j f(x_0 + c_j h, g_j)\end{aligned}$$

- On choisit de travailler avec des quantités plus petites afin de réduire les problèmes d'erreurs d'arrondis :

$$z_i = g_i - y_0$$

ce qui permet de réécrire la première équation :

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

- Ainsi, lorsque les variables z_1, \dots, z_s sont connues, la deuxième équation devient explicite pour y_1 . Une application directe de ce raisonnement nécessite s évaluations de fonctions additionnelles.

Réécriture de la méthode

On peut éviter ces calculs dans le cas où la matrice $A = (a_{ij})$ n'est pas singulière

$$\begin{pmatrix} z_1 \\ \vdots \\ z_s \end{pmatrix} = A \begin{pmatrix} hf(x_0 + c_1 h, y_0 + z_1) \\ \vdots \\ hf(x_0 + c_s h, y_0 + z_s) \end{pmatrix}$$

Et ainsi

$$y_1 = y_0 + \sum_{i=1}^s d_i z_i, \quad (d_1, \dots, d_s) = (b_1, \dots, b_s) A^{-1}$$

Dans le cas qui nous intéresse, c'est-à-dire $s = 3$, $d = (0, 0, 1)$, puisque $b_i = a_{si} \forall i$.

Algorithme de Newton modifié

Méthode itérative conduit à chaque itération à l'inversion de :

$$\begin{pmatrix} I - ha_{11} \frac{\partial f}{\partial y}(x_0 + c_1 h, y_0 + z_1) & \cdots & -ha_{1s} \frac{\partial f}{\partial y}(x_0 + c_s h, y_0 + z_s) \\ \vdots & & \vdots \\ -ha_{s1} \frac{\partial f}{\partial y}(x_0 + c_1 h, y_0 + z_1) & \cdots & I - ha_{ss} \frac{\partial f}{\partial y}(x_0 + c_s h, y_0 + z_s) \end{pmatrix}$$

Approximation de la Jacobienne $J \approx \frac{\partial f}{\partial y}(x_0, y_0)$

$$(I - hA \otimes J) \Delta Z^k = -Z^k + h(A \otimes I) F(Z^k)$$

$$Z^{k+1} = Z^k + \Delta Z^k$$

$Z^k = (z_1^k, \dots, z_s^k)^T$ est la k ème approximation, $\Delta Z^k = (\Delta z_1^k, \dots, \Delta z_s^k)^T$ les incréments, $F(Z^k) = (f(x_0 + c_1 h, y_0 + z_1^k), \dots, f(x_0 + c_s h, y_0 + z_s^k))^T$

→ Chaque itération nécessite s évaluations de f , et la résolution d'un système linéaire de taille ns . Par contre, la matrice $(I - hA \otimes J)$ est la même pour toutes les itérations. Sa décomposition LU n'est donc réalisée qu'une seule fois.

Approximation initiale

Comme la solution exacte vérifie $z_i = \mathcal{O}(h)$, le choix le plus simple est :

$$z_i^0 = 0 \quad i = 1, \dots, s$$

On peut faire des choix plus efficaces. Si q est le polynôme d'interpolation de degré s définit par : $q(0) = 0$, $q(c_i) = z_i$, $i = 1 \dots, s$, alors la condition initiale :

$$\begin{aligned} z_i^0 &= q(1 + wc_i) + y_0 - y_1 \quad i = 1, \dots, s \\ w &= \frac{h_{new}}{h_{old}} \end{aligned}$$

donne une convergence numérique plus rapide

Estimation d'erreur et critère d'arrêt

Comme la convergence est linéaire, on a, en espérant que $\Theta < 1$:

$$\|\Delta Z^{k+1}\| \leq \Theta \|\Delta Z^k\|$$

Si on applique l'inégalité triangulaire au développement (Z^* est la solution exacte)
 $Z^{k+1} - Z^* = (Z^{k+1} - Z^{k+2}) + (Z^{k+2} - Z^{k+3}) + \dots$, on obtient l'estimation d'erreur :

$$\|Z^{k+1} - Z^*\| \leq \frac{\Theta}{1 - \Theta} \|\Delta Z^k\|$$

Le taux de convergence peut être estimé à partir de quantités calculées :

$$\Theta_k = \frac{\|\Delta Z^k\|}{\|\Delta Z^{k-1}\|}$$

L'erreur numérique ne doit pas être supérieure à l'erreur de discrétisation locale proche de *Tol*. Les itérations sont donc arrêtées quand :

$$\eta_k \|\Delta Z^k\| \leq \kappa \cdot Tol \text{ avec } \eta_k = \frac{\Theta_k}{1 - \Theta_k}$$

Cela ne peut s'appliquer qu'après 2 itérations de la méthode.

Estimation d'erreur et critère d'arrêt

Il reste encore à choisir le paramètre κ . Des expériences numériques montrent une meilleure efficacité du code pour des valeurs de κ autour de 10^{-1} ou 10^{-2} .

De même, il semble que le code soit plus efficace quand le nombre maximum d'itérations k_{max} est élevé ($k_{max} = 7$ ou 10). Durant ces itérations, le calcul est interrompu et redémarré avec un pas plus petit (par exemple avec $h := h/2$) dans le cas où on se trouve dans une des situations suivantes :

- $\Theta_k \geq 1$ pour un k ,
- Pour quelques k ,

$$\frac{\Theta_k^{k_{max}-k}}{1 - \Theta_k} \|\Delta Z^k\| > \kappa.Tol$$

Si une seule itération de Newton est suffisante pour satisfaire le critère d'arrêt ou si le dernier $\Theta_k \leq 10^{-3}$, le jacobien n'est pas recalculé au pas suivant.

Plan de la présentation

- 1 Rappel sur les limitations des méthodes classiques
 - La notion de A-stabilité
 - Les méthodes de Runge-Kutta implicite
- 2 **La méthode RadauIIA**
 - La méthode RadauIIA
 - Reformulation du système et méthode de Newton simplifiée
 - **Résolution du système linéaire**
 - Contrôle du pas
- 3 **Résolution de l'équation de la chaleur**
 - Méthodes implicites
 - Implémentation
 - Méthodes explicites

Résolution efficace du système linéaire

La résolution se fait en exploitant la structure particulière de la matrice $I - hA \otimes J$.
L'idée est de multiplier par $(hA)^{-1} \otimes I$ et de transformer A^{-1} en une matrice simple (diagonal par bloc, triangulaire ...):

$$T^{-1}A^{-1}T = \Lambda$$

Si on considère le changement de variables $W^k = (T^{-1} \otimes I)Z^k$, on obtient :

$$(h^{-1}\Lambda \otimes I - I \otimes J)\Delta W^k = -h^{-1}(\Lambda \otimes I)W^k + (T^{-1} \otimes I)F((T \otimes I)W^k)$$

$$W^{k+1} = W^k + \Delta W^k$$

Résolution efficace du système linéaire

Si on suppose que A^{-1} a une valeur propre réel $\hat{\gamma}$ et une paire de valeurs propres complexes conjuguées $\hat{\alpha} \pm i\hat{\beta}$ (ce qui est le cas pour Radau IIA), la matrice se réécrit :

$$\begin{pmatrix} \gamma I - J & 0 & 0 \\ 0 & \alpha I - J & -\beta I \\ 0 & \beta I & \alpha I - J \end{pmatrix}$$

avec $\gamma = h^{-1}\hat{\gamma}$, $\alpha = h^{-1}\hat{\alpha}$, $\beta = h^{-1}\hat{\beta}$.

Ainsi, le système peut se décomposer en deux systèmes linéaires de taille n et $2n$.
D'autres idées sont possibles pour exploiter la structure particulière de la matrice $2n \times 2n$.

Plan de la présentation

- 1 Rappel sur les limitations des méthodes classiques
 - La notion de A-stabilité
 - Les méthodes de Runge-Kutta implicite
- 2 **La méthode RadauIIA**
 - La méthode RadauIIA
 - Reformulation du système et méthode de Newton simplifiée
 - Résolution du système linéaire
 - **Contrôle du pas**
- 3 Résolution de l'équation de la chaleur
 - Méthodes implicites
 - Implémentation
 - Méthodes explicites

Calcul de l'erreur pour le contrôle

Comme la méthode est d'ordre optimale, on ne peut pas se baser sur une autre plus précise pour estimer l'erreur. Donc, on considère une méthode d'ordre moins élevé de la forme :

$$\hat{y}_1 = y_0 + h(\hat{b}_0 f(x_0, y_0) + \sum_{i=1}^3 \hat{b}_i f(x_0 + c_i h, g_i))$$

où les g_1, g_2, g_3 sont les valeurs obtenues pour la méthode de Radau IIA, et $\hat{b}_0 \neq 0$.
On peut écrire la différence entre les solutions des deux méthodes ($\hat{b}_0 = \gamma_0 = \hat{\gamma}^{-1}$) :

$$\hat{y}_1 - y_1 = \gamma_0 h f(x_0, y_0) + \sum_{i=1}^3 (\hat{b}_i - b_i) h f(x_0 + c_i h, g_i) = \gamma_0 h f(x_0, y_0) + e_1 z_1 + e_2 z_2 + e_3 z_3,$$

dont on se sert pour l'évaluation de l'erreur :

$$err = (I - h\gamma_0 J)^{-1}(\hat{y}_1 - y_1)$$

Dans le cas du premier pas, et après chaque pas rejeté pour lesquels $\|err\| > 1$, l'erreur est calculé par :

$$\widetilde{err} = (I - h\gamma_0 J)^{-1}(\gamma_0 h f(x_0, y_0 + err) + e_1 z_1 + e_2 z_2 + e_3 z_3)$$

Contrôle standard

Les expressions se comportent en $\mathcal{O}(h^4)$, la prédiction du pas s'écrit donc :

$$h_{new} = fac.h_{old} \cdot \|err\|^{-1/4}$$

avec $\|err\| = \sqrt{\frac{1}{n} \sum_{i=1}^n (\frac{err_i}{sc_i})^2}$ et $sc_i = Atoi + \max(|y_{0i}|, |y_{1i}|) \cdot Rtoi$. Le coefficient *fac* dépend du nombre d'itérations de Newton *Newt*
 $fac = 0.9 \times (2k_{max} + 1) / (2k_{max} + Newt)$

Pour limiter le nombre de décomposition LU de la matrice, si le jacobien n'est pas recalculé et si $c_1 h_{old} < h_{new} < c_2 h_{old}$ avec $c_1 = 1.0$ et $c_2 = 1.2$, alors on conserve h_{old} pour le pas suivant.

Contrôle prédictif

Dans le cas de systèmes très raides, la décroissance du pas de temps peut devoir être extrêmement rapide alors que la précédente est limitée par le coefficient *fac*.

Si on note err_{n+1} l'erreur donnée et si h_n correspond au pas du calcul de l'étape n , la prédiction du pas est issue de la formule asymptotique :

$$\|err_{n+1}\| = C_n h_n^4$$

L'évaluation de h_{new} est basé sur l'hypothèse que $C_{n+1} \approx C_n$. Un meilleur modèle peut être obtenu en considérant que $\log C_n$ est une fonction linéaire de n , et donc que $\log C_{n+1} - \log C_n$ est constant ou encore :

$$C_{n+1}/C_n \approx C_n/C_{n-1}$$

On obtient ainsi une nouvelle estimation du pas :

$$h_{new} = fac.h_n \left(\frac{1}{\|err_{n+1}\|} \right)^{1/4} \cdot \frac{h_n}{h_{n+1}} \left(\frac{\|err_n\|}{\|err_{n+1}\|} \right)^{1/4}$$

Dans le code, le nouveau pas correspond au minimum des résultats obtenus précédemment

Plan de la présentation

- 1 Rappel sur les limitations des méthodes classiques
 - La notion de A-stabilité
 - Les méthodes de Runge-Kutta implicite
- 2 La méthode RadauIIA
 - La méthode RadauIIA
 - Reformulation du système et méthode de Newton simplifiée
 - Résolution du système linéaire
 - Contrôle du pas
- 3 Résolution de l'équation de la chaleur
 - Méthodes implicites
 - Implémentation
 - Méthodes explicites

Discrétisation des équations de la chaleur

A priori, on a besoin d'une méthode L-stable, d'ordre 2 au moins.

Méthode SDIRK de Ropp et Shadid : 2 systèmes linéaires à résoudre à chaque pas, mais avec la même matrice.

$$\begin{aligned}k_1 &= \Delta(u_n + \gamma h k_1), \\k_2 &= \Delta(u_n + (1 - 2\gamma h)k_1 + \gamma h k_2), \\ \text{avec : } \gamma &= 1 + 1/\sqrt{2}; \\ \text{puis : } u_{n+1} &= u_n + \frac{h}{2}(k_1 + k_2).\end{aligned}$$

Discrétisation des équations de la chaleur

A priori, on a besoin d'une méthode L-stable, d'ordre 2 au moins.

Méthode SDIRK de Ropp et Shadid : 2 systèmes linéaires à résoudre à chaque pas, mais avec la même matrice.

$$\begin{aligned}k_1 &= \Delta(u_n + \gamma h k_1), \\k_2 &= \Delta(u_n + (1 - 2\gamma h)k_1 + \gamma h k_2), \\ \text{avec : } \gamma &= 1 + 1/\sqrt{2}; \\ \text{puis : } u_{n+1} &= u_n + \frac{h}{2}(k_1 + k_2).\end{aligned}$$

Discrétisation des équations de la chaleur

A priori, on a besoin d'une méthode L-stable, d'ordre 2 au moins.

Méthode SDIRK de Ropp et Shadid : 2 systèmes linéaires à résoudre à chaque pas, mais avec la même matrice.

$$\begin{aligned}k_1 &= \Delta(u_n + \gamma h k_1), \\k_2 &= \Delta(u_n + (1 - 2\gamma h)k_1 + \gamma h k_2), \\ \text{avec : } \gamma &= 1 + 1/\sqrt{2}; \\ \text{puis : } u_{n+1} &= u_n + \frac{h}{2}(k_1 + k_2).\end{aligned}$$

Discrétisation des équations de la chaleur

A priori, on a besoin d'une méthode L-stable, d'ordre 2 au moins.

Méthode SDIRK de Ropp et Shadid : 2 systèmes linéaires à résoudre à chaque pas, mais avec la même matrice.

$$\begin{aligned}k_1 &= \Delta(u_n + \gamma h k_1), \\k_2 &= \Delta(u_n + (1 - 2\gamma h)k_1 + \gamma h k_2), \\ \text{avec : } \gamma &= 1 + 1/\sqrt{2}; \\ \text{puis : } u_{n+1} &= u_n + \frac{h}{2}(k_1 + k_2).\end{aligned}$$

Résolution des systèmes linéaires

Dimension 2 : 10 systèmes de 150 000 inconnues par pas... Dimension 3 ? 10 systèmes de quelques millions d'inconnues.

- **dimension 2** : SuperLU ou un de ses avatars : imbatables : temps de factorisation "important" mais on ne factorise qu'une fois. Temps de résolution *négligeable*.
- **dimension 3** : SuperLU, MUMPS inutilisables sur une machine raisonnable. Meilleure méthode : **multigrille algébrique (Y. Notay)** : peu de mémoire ; 1,5 secondes pour 1 million d'inconnues en 3d, assez bien parallélisée.

Résolution des systèmes linéaires

Dimension 2 : 10 systèmes de 150 000 inconnues par pas... Dimension 3 ? 10 systèmes de quelques millions d'inconnues.

- **dimension 2** : SuperLU ou un de ses avatars : imbatables : temps de factorisation "important" mais on ne factorise qu'une fois. Temps de résolution *négligeable*.
- **dimension 3** : SuperLU, MUMPS inutilisables sur une machine raisonnable. Meilleure méthode : **multigrille algébrique (Y. Notay)** : peu de mémoire ; 1,5 secondes pour 1 million d'inconnues en 3d, assez bien parallélisée.

Résolution des systèmes linéaires

Dimension 2 : 10 systèmes de 150 000 inconnues par pas... Dimension 3 ? 10 systèmes de quelques millions d'inconnues.

- **dimension 2** : SuperLU ou un de ses avatars : imbatables : temps de factorisation "important" mais on ne factorise qu'une fois. Temps de résolution *négligeable*.
- **dimension 3** : SuperLU, MUMPS inutilisables sur une machine raisonnable.
Meilleure méthode : **multigrille algébrique (Y. Notay)** : peu de mémoire ; 1,5 secondes pour 1 million d'inconnues en 3d, assez bien parallélisée.

Résolution des systèmes linéaires

Dimension 2 : 10 systèmes de 150 000 inconnues par pas... Dimension 3 ? 10 systèmes de quelques millions d'inconnues.

- **dimension 2** : SuperLU ou un de ses avatars : imbatables : temps de factorisation "important" mais on ne factorise qu'une fois. Temps de résolution *négligeable*.
- **dimension 3** : SuperLU, MUMPS inutilisables sur une machine raisonnable. Meilleure méthode : **multigrille algébrique (Y. Notay)** : peu de mémoire ; 1,5 secondes pour 1 million d'inconnues en 3d, assez bien parallélisée.

Résolution des systèmes linéaires

Dimension 2 : 10 systèmes de 150 000 inconnues par pas... Dimension 3 ? 10 systèmes de quelques millions d'inconnues.

- **dimension 2** : SuperLU ou un de ses avatars : imbatables : temps de factorisation "important" mais on ne factorise qu'une fois. Temps de résolution *négligeable*.
- **dimension 3** : SuperLU, MUMPS inutilisables sur une machine raisonnable. Meilleure méthode : **multigrille algébrique (Y. Notay)** : peu de mémoire ; 1,5 secondes pour 1 million d'inconnues en 3d, assez bien parallélisée.

Plan de la présentation

- 1 Rappel sur les limitations des méthodes classiques
 - La notion de A-stabilité
 - Les méthodes de Runge-Kutta implicite
- 2 La méthode RadauIIA
 - La méthode RadauIIA
 - Reformulation du système et méthode de Newton simplifiée
 - Résolution du système linéaire
 - Contrôle du pas
- 3 Résolution de l'équation de la chaleur
 - Méthodes implicites
 - **Implémentation**
 - Méthodes explicites

Implémentation

Discrétisation spatiale dans la modélisation des accidents vasculaires cérébraux : 1 pixel = 1 volume fini... 150 000 volumes en dimension 2. Plusieurs millions en 3d (on part d'IRMs).

Code Zebre :

<http://math.univ-lyon1.fr/~tdumont/zebre>

- 1d, 2d, 3d, différences finies, éléments finis, volumes finis (structurés),
- "tous" les schémas.
- multithreadé (Réaction et diffusion).
- C++ + bibliothèques fortran,
- Blitz
- Boost (threads...)

Implémentation

Discrétisation spatiale dans la modélisation des accidents vasculaires cérébraux : 1 pixel = 1 volume fini... 150 000 volumes en dimension 2. Plusieurs millions en 3d (on part d'IRMs).

Code Zebre :

<http://math.univ-lyon1.fr/~tdumont/zebre>

- 1d, 2d, 3d, différences finies, éléments finis, volumes finis (structurés),
- “tous” les schémas.
- multithreadé (Réaction et diffusion).
- C++ + bibliothèques fortran,
- Blitz
- Boost (threads...)

Implémentation

Discrétisation spatiale dans la modélisation des accidents vasculaires cérébraux : 1 pixel = 1 volume fini... 150 000 volumes en dimension 2. Plusieurs millions en 3d (on part d'IRMs).

Code Zebre :

<http://math.univ-lyon1.fr/~tdumont/zebre>

- 1d, 2d, 3d, différences finies, éléments finis, volumes finis (structurés),
- “tous” les schémas.
- multithreadé (Réaction et diffusion).
- C++ + bibliothèques fortran,
- Blitz
- Boost (threads...)

Implémentation

Discrétisation spatiale dans la modélisation des accidents vasculaires cérébraux : 1 pixel = 1 volume fini... 150 000 volumes en dimension 2. Plusieurs millions en 3d (on part d'IRMs).

Code Zebre :

<http://math.univ-lyon1.fr/~tdumont/zebre>

- 1d, 2d, 3d, différences finies, éléments finis, volumes finis (structurés),
- “tous” les schémas.
- multithreadé (Réaction et diffusion).
- C++ + bibliothèques fortran,
- Blitz
- Boost (threads...)

Implémentation

Discrétisation spatiale dans la modélisation des accidents vasculaires cérébraux : 1 pixel = 1 volume fini... 150 000 volumes en dimension 2. Plusieurs millions en 3d (on part d'IRMs).

Code Zebre :

<http://math.univ-lyon1.fr/~tdumont/zebre>

- 1d, 2d, 3d, différences finies, éléments finis, volumes finis (structurés),
- “tous” les schémas.
- multithreadé (Réaction et diffusion).
- C++ + bibliothèques fortran,
- Blitz
- Boost (threads...)

Implémentation

Discrétisation spatiale dans la modélisation des accidents vasculaires cérébraux : 1 pixel = 1 volume fini... 150 000 volumes en dimension 2. Plusieurs millions en 3d (on part d'IRMs).

Code Zebre :

<http://math.univ-lyon1.fr/~tdumont/zebre>

- 1d, 2d, 3d, différences finies, éléments finis, volumes finis (structurés),
- “tous” les schémas.
- multithreadé (Réaction et diffusion).

- C++ + bibliothèques fortran,
- Blitz
- Boost (threads...)

Plan de la présentation

- 1 Rappel sur les limitations des méthodes classiques
 - La notion de A-stabilité
 - Les méthodes de Runge-Kutta implicite
- 2 La méthode RadauIIA
 - La méthode RadauIIA
 - Reformulation du système et méthode de Newton simplifiée
 - Résolution du système linéaire
 - Contrôle du pas
- 3 Résolution de l'équation de la chaleur
 - Méthodes implicites
 - Implémentation
 - Méthodes explicites

Quoi de neuf ? les méthodes explicites !

On vous aurait menti ?

Pour atteindre la dimension 3 :

- paralléliser (mpi) : multigrille algébrique... ok réactions ?
- et/ou changer de méthode pour la diffusion.

Diffusion relativement faible \Rightarrow équations de la chaleur pas très raides.

Quoi de neuf ? les méthodes explicites !

On vous aurait menti ?

Pour atteindre la dimension 3 :

- paralléliser (mpi) : multigrille algébrique... ok réactions ?
- et/ou changer de méthode pour la diffusion.

Diffusion relativement faible \Rightarrow équations de la chaleur pas très raides.

Quoi de neuf ? les méthodes explicites !

On vous aurait menti ?

Pour atteindre la dimension 3 :

- paralléliser (mpi) : multigrille algébrique... ok réactions ?
- et/ou changer de méthode pour la diffusion.

Diffusion relativement faible \Rightarrow équations de la chaleur pas *très* raides.

Quoi de neuf ? les méthodes explicites !

On vous aurait menti ?

Pour atteindre la dimension 3 :

- paralléliser (mpi) : multigrille algébrique... ok réactions ?
- et/ou changer de méthode pour la diffusion.

Diffusion relativement faible => équations de la chaleur pas *très* raides.

Quoi de neuf ? les méthodes explicites !

On vous aurait menti ?

Pour atteindre la dimension 3 :

- paralléliser (mpi) : multigrille algébrique... ok réactions ?
- et/ou changer de méthode pour la diffusion.

Diffusion relativement faible => équations de la chaleur pas *très* raides.

Quoi de neuf ? les méthodes explicites !

On vous aurait menti ?

Pour atteindre la dimension 3 :

- paralléliser (mpi) : multigrille algébrique... ok réactions ?
- et/ou changer de méthode pour la diffusion.

Diffusion relativement faible \Rightarrow équations de la chaleur pas *très* raides.

Quoi de neuf ? les méthodes explicites !

Idée : fabriquer une méthode (de Runge–Kutta) explicite ayant le plus grand domaine de stabilité possible (du côté des réels négatifs).

- vieille idée : Guillou et Soulé ('60),
- revisitée complètement par A. Abdulle et Cie : méthodes ROCK2 et ROCK4

Quoi de neuf ? les méthodes explicites !

Idée : fabriquer une méthode (de Runge–Kutta) explicite ayant le plus grand domaine de stabilité possible (du côté des réels négatifs).

- vieille idée : Guillou et Soulé ('60),
- revisitée complètement par A. Abdulle et Cie : méthodes ROCK2 et ROCK4

Quoi de neuf ? les méthodes explicites !

Idée : fabriquer une méthode (de Runge–Kutta) explicite ayant le plus grand domaine de stabilité possible (du coté des réels négatifs).

- vieille idée : Guillou et Soulé ('60),
- revisitée complètement par A. Abdulle et Cie : méthodes ROCK2 et ROCK4

Méthodes Rock

Méthode explicite $\Rightarrow R(z)$ est un polynôme.

A l'ordre 1, il suffit que $R(z) = 1 + z + \dots \Rightarrow$

$$R(z) = T_n(1 + z/n^2) \text{ convient.}$$

intervalle de stabilité $-2n^2 \leq z \leq 0$.

Ordre supérieur ?

Lebedev, Medovikov, Abdulle (ROCK) :

$R_n(z) = w_p(z)P_{n-p}(z)$ avec :

- les polynômes $P_k(z)$ orthogonaux par rapport à $\frac{w_p(z)}{\sqrt{1-z^2}}$.
- $w_p(z)$ choisit pour que la méthode soit d'ordre p .

Méthodes Rock

Méthode explicite $\Rightarrow R(z)$ est un polynôme.
A l'ordre 1, il suffit que $R(z) = 1 + z + \dots \Rightarrow$

$$R(z) = T_n(1 + z/n^2) \text{ convient.}$$

intervalle de stabilité $-2n^2 \leq z \leq 0$.

Ordre supérieur ?

Lebedev, Medovikov, Abdulle (ROCK) :

$R_n(z) = w_p(z)P_{n-p}(z)$ avec :

- les polynômes $P_k(z)$ orthogonaux par rapport à $\frac{w_p(z)}{\sqrt{1-z^2}}$.
- $w_p(z)$ choisit pour que la méthode soit d'ordre p .

Méthodes Rock

Méthode explicite $\Rightarrow R(z)$ est un polynôme.
A l'ordre 1, il suffit que $R(z) = 1 + z + \dots \Rightarrow$

$$R(z) = T_n(1 + z/n^2) \text{ convient.}$$

intervalle de stabilité $-2n^2 \leq z \leq 0$.

Ordre supérieur ?

Lebedev, Medovikov, Abdulle (ROCK) :

$R_n(z) = w_p(z)P_{n-p}(z)$ avec :

- les polynômes $P_k(z)$ orthogonaux par rapport à $\frac{w_p(z)}{\sqrt{1-z^2}}$.
- $w_p(z)$ choisit pour que la méthode soit d'ordre p .

Méthodes Rock

Méthode explicite $\Rightarrow R(z)$ est un polynôme.
A l'ordre 1, il suffit que $R(z) = 1 + z + \dots \Rightarrow$

$$R(z) = T_n(1 + z/n^2) \text{ convient.}$$

intervalle de stabilité $-2n^2 \leq z \leq 0$.

Ordre supérieur ?

Lebedev, Medovikov, Abdulle (ROCK) :

$R_n(z) = w_p(z)P_{n-p}(z)$ avec :

- les polynômes $P_k(z)$ orthogonaux par rapport à $\frac{w_p(z)}{\sqrt{1-z^2}}$.
- $w_p(z)$ choisit pour que la méthode soit d'ordre p .

Méthodes Rock

Méthode explicite $\Rightarrow R(z)$ est un polynôme.
A l'ordre 1, il suffit que $R(z) = 1 + z + \dots \Rightarrow$

$$R(z) = T_n(1 + z/n^2) \text{ convient.}$$

intervalle de stabilité $-2n^2 \leq z \leq 0$.

Ordre supérieur ?

Lebedev, Medovikov, Abdulle (ROCK) :

$R_n(z) = w_p(z)P_{n-p}(z)$ avec :

- les polynômes $P_k(z)$ orthogonaux par rapport à $\frac{w_p(z)}{\sqrt{1-z^2}}$.
- $w_p(z)$ choisit pour que la méthode soit d'ordre p .

Méthodes Rock

Méthode explicite $\Rightarrow R(z)$ est un polynôme.
A l'ordre 1, il suffit que $R(z) = 1 + z + \dots \Rightarrow$

$$R(z) = T_n(1 + z/n^2) \text{ convient.}$$

intervalle de stabilité $-2n^2 \leq z \leq 0$.

Ordre supérieur ?

Lebedev, Medovikov, Abdulle (ROCK) :

$R_n(z) = w_p(z)P_{n-p}(z)$ avec :

- les polynômes $P_k(z)$ orthogonaux par rapport à $\frac{w_p(z)}{\sqrt{1-z^2}}$.
- $w_p(z)$ choisit pour que la méthode soit d'ordre p .

Méthodes Rock

Ces méthodes peuvent être vues comme des Runge–Kutta (explicites).

Un théorème de Bernstein (1930) implique que ces méthodes sont *presque* optimales (intervalle de stabilité en p^2).

- à l'ordre 2 : assez facile : l'étude linéaire suffit.
- à l'ordre 4 : difficile ! composition de 2 méthodes RK... groupe de Butcher etc...
Note : il existe une version **stochastique** (SROCK4) de ROCK4.
- programmation : utiliser la formule de récurrence à 3 termes.
- multiples raffinements : voir le code de Abdulle et les publications.

Méthodes Rock

Ces méthodes peuvent être vues comme des Runge–Kutta (explicites).
Un théorème de Bernstein (1930) implique que ces méthodes sont *presque* optimales (intervalle de stabilité en p^2).

- à l'ordre 2 : assez facile : l'étude linéaire suffit.
- à l'ordre 4 : difficile ! composition de 2 méthodes RK... groupe de Butcher etc...
Note : il existe une version **stochastique** (SROCK4) de ROCK4.
- programmation : utiliser la formule de récurrence à 3 termes.
- multiples raffinements : voir le code de Abdulle et les publications.

Méthodes Rock

Ces méthodes peuvent être vues comme des Runge–Kutta (explicites).
Un théorème de Bernstein (1930) implique que ces méthodes sont *presque* optimales (intervalle de stabilité en p^2).

- à l'ordre 2 : assez facile : l'étude linéaire suffit.
- à l'ordre 4 : difficile ! composition de 2 méthodes RK... groupe de Butcher etc...
Note : il existe une version **stochastique** (SROCK4) de ROCK4.
- programmation : utiliser la formule de récurrence à 3 termes.
- multiples raffinements : voir le code de Abdulle et les publications.

Méthodes Rock

Ces méthodes peuvent être vues comme des Runge–Kutta (explicites).
Un théorème de Bernstein (1930) implique que ces méthodes sont *presque* optimales (intervalle de stabilité en p^2).

- à l'ordre 2 : assez facile : l'étude linéaire suffit.
- à l'ordre 4 : difficile ! composition de 2 méthodes RK... groupe de Butcher etc...
Note : il existe une version **stochastique** (SROCK4) de ROCK4.
- programmation : utiliser la formule de récurrence à 3 termes.
- multiples raffinements : voir le code de Abdulle et les publications.

Méthodes Rock

- Equation de la chaleur : **uniquement des produits matrice vecteur**.
- Il faut estimer le rayon spectral de la matrice du Laplacien (disques de Gershgorin).