

*Simulations numériques :
méthodes de différences finies*

ED A&A IdF, module post-M2

Simulations numériques et calculs hautes performances

Éric Buchlin

`eric.buchlin@ias.u-psud.fr`

Janvier 2017

Que veut-on faire ?

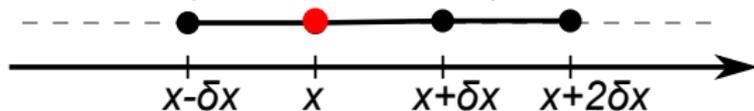
Résoudre des équations différentielles ordinaires (ODE / Ordinary Differential Equation) et, le plus souvent en physique, des équations aux dérivées partielles (PDE, Partial Differential Equation), avec différentes *conditions initiales* (et/ou aux bords).

Que veut-on faire ?

Résoudre des équations différentielles ordinaires (ODE / Ordinary Differential Equation) et, le plus souvent en physique, des équations aux dérivées partielles (PDE, Partial Differential Equation), avec différentes *conditions initiales* (et/ou aux bords).

Comment ?

Après *discrétisation* de l'espace des variables (grille de points), on utilise des *approximations des dérivées*, obtenues grâce à des *développements limités* des variables sur un certain nombre de *points voisins* du point courant.



Approximation de dérivées par des différences finies

1. *Approximation de dérivées par des différences finies*

2. *Équations différentielles ordinaires avec condition initiale*

3. *Équations aux dérivées partielles*

4. *Exemple : un tube de flux turbulent*

Discrétisation

La variable x est discrétisée par :

$$x_{i+1} \equiv x_i + h_i \quad (1)$$

et on définit la version discrétisée de la fonction dont on veut calculer les dérivées :

$$u_i \equiv u(x_i) \quad (2)$$

Discrétisation

La variable x est discrétisée par :

$$x_{i+1} \equiv x_i + h_i \quad (1)$$

et on définit la version discrétisée de la fonction dont on veut calculer les dérivées :

$$u_i \equiv u(x_i) \quad (2)$$

Notes :

- ▶ u peut être un vecteur.
- ▶ Pour simplifier, dans la suite on prendra $h_i = \delta x$ (grille régulière)
- ▶ On doit d'abord choisir l'ensemble des x_j qu'on veut utiliser pour calculer chaque dérivée en x_i (“*stencil*”).
- ▶ La dérivée de l'équation différentielle est obtenue par combinaison linéaire de développements limités, à un certain *ordre* de précision.

Approximation numérique de la dérivée

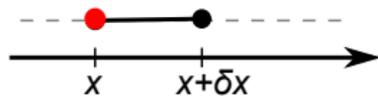
Le développement limité (DL) d'ordre 1

$$u_{i+1} = u(x_i + \delta x) = u_i + \delta x u'_i + \mathcal{O}(\delta x^2) \quad (3)$$

permet d'écrire $u'_i = \left(\frac{du}{dt}\right)_{x=x_i}$ sous la forme

$$u'_i = \frac{1}{\delta x} (u_{i+1} - u_i) + \mathcal{O}(\delta x) \quad (4)$$

(stencil = $\{i, i+1\}$, schéma «en avant»).



Approximation numérique de la dérivée

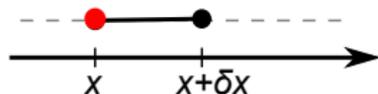
Le développement limité (DL) d'ordre 1

$$u_{i+1} = u(x_i + \delta x) = u_i + \delta x u'_i + \mathcal{O}(\delta x^2) \quad (3)$$

permet d'écrire $u'_i = \left(\frac{du}{dx}\right)_{x=x_i}$ sous la forme

$$u'_i = \frac{1}{\delta x} (u_{i+1} - u_i) + \mathcal{O}(\delta x) \quad (4)$$

(stencil = $\{i, i+1\}$, schéma «en avant»).

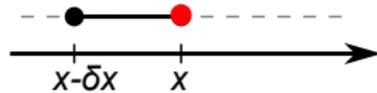


De même, en partant de

$$u_{i-1} = u(x_i - \delta x) = u_i - \delta x u'_i + \mathcal{O}(\delta x^2) \quad (5)$$

on obtient
$$u'_i = \frac{1}{\delta x} (u_i - u_{i-1}) + \mathcal{O}(\delta x) \quad (6)$$

(stencil = $\{i-1, i\}$, schéma «en arrière»).



Approximation numérique de la dérivée

Pour avoir une meilleure précision, on utilise des *DL d'ordre supérieur*.

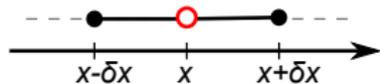
Par exemple en écrivant les DL

$$u_{i+1} = u(x_i + \delta x) = u_i + \delta x u'_i + \frac{\delta x^2}{2} u''_i + \mathcal{O}(\delta x^3) \quad (7)$$

$$u_{i-1} = u(x_i - \delta x) = u_i - \delta x u'_i + \frac{\delta x^2}{2} u''_i + \mathcal{O}(\delta x^3) \quad (8)$$

(9)

on voit qu'on peut obtenir u'_i par



$$u'_i = \frac{1}{2\delta x} (u_{i+1} - u_{i-1}) + \mathcal{O}(\delta x^2) \quad (\text{schéma centré}) \quad (10)$$

Approximation numérique de la dérivée

Pour avoir une meilleure précision, on utilise des *DL d'ordre supérieur*.

Par exemple en écrivant les DL

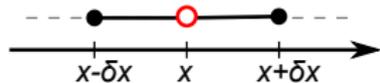
$$u_{i+1} = u(x_i + \delta x) = u_i + \delta x u'_i + \frac{\delta x^2}{2} u''_i + \mathcal{O}(\delta x^3) \quad (7)$$

$$u_{i-1} = u(x_i - \delta x) = u_i - \delta x u'_i + \frac{\delta x^2}{2} u''_i + \mathcal{O}(\delta x^3) \quad (8)$$

(9)

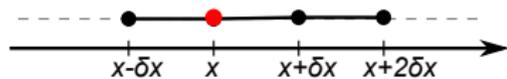
on voit qu'on peut obtenir u'_i par

$$u'_i = \frac{1}{2\delta x} (u_{i+1} - u_{i-1}) + \mathcal{O}(\delta x^2) \quad (\text{schéma centré}) \quad (10)$$



De même, à l'ordre 3, on a par exemple

$$u'_i = \frac{1}{6\delta x} (-2u_{i-1} - 3u_i + 6u_{i+1} - u_{i+2}) + \mathcal{O}(\delta x^3) \quad (11)$$



Dérivées d'ordre supérieur et dérivées croisées

On utilise le même principe que pour les dérivées premières.

Pour les *dérivées d'ordre supérieur*, on a par exemple :

$$u_i'' = \frac{1}{\delta x^2} (u_{i-1} - 2u_i + u_{i+1}) + \mathcal{O}(\delta x) \quad (\text{centré}), \text{ ou} \quad (12)$$

$$= \frac{1}{\delta x^2} (u_i - 2u_{i+1} + u_{i+2}) + \mathcal{O}(\delta x) \quad (\text{en avant}), \text{ ou...} \quad (13)$$

Dérivées d'ordre supérieur et dérivées croisées

On utilise le même principe que pour les dérivées premières.

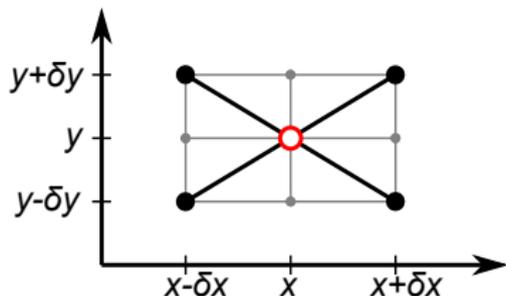
Pour les *dérivées d'ordre supérieur*, on a par exemple :

$$u_i'' = \frac{1}{\delta x^2} (u_{i-1} - 2u_i + u_{i+1}) + \mathcal{O}(\delta x) \quad (\text{centré}), \text{ ou} \quad (12)$$

$$= \frac{1}{\delta x^2} (u_i - 2u_{i+1} + u_{i+2}) + \mathcal{O}(\delta x) \quad (\text{en avant}), \text{ ou...} \quad (13)$$

Pour $u_{i,j} = u(x_i, y_j)$, on peut calculer la *dérivée croisée* par exemple par

$$\left(\frac{\partial^2 u}{\partial x \partial y} \right) = \frac{u_{i-1,j-1} - u_{i-1,j+1} - u_{i+1,j-1} + u_{i+1,j+1}}{4 \delta x \delta y} + \mathcal{O}(\delta x^2, \delta y^2) \quad (14)$$



Équations différentielles ordinaires avec condition initiale

1. Approximation de dérivées par des différences finies

2. Équations différentielles ordinaires avec condition initiale

3. Équations aux dérivées partielles

4. Exemple : un tube de flux turbulent

Le problème à résoudre : le problème de Cauchy

Trouver la fonction $u(t)$ définie sur l'intervalle $[a, b]$ telle que

$$u'(t) = f(t, u(t)) \quad \text{et} \quad u(a) \text{ donné} \quad (15)$$

Notes :

- ▶ Le théorème de Cauchy-Lipschitz donne des conditions d'existence et d'unicité ($f(t, u)$ continue, lipschitzienne par rapport à u ...)
- ▶ Une ODE d'ordre supérieur peut se ramener à une ODE du premier ordre en introduisant des variables supplémentaires.
- ▶ Après discrétisation ($t_{n+1} = t_n + \delta t$), on veut obtenir une bonne *approximation de u_n* de $u(t_n)$.
- ▶ Pour cela on calcule récursivement u_{n+1} une fois connues les valeurs précédentes $\{u_i \mid i \leq n\}$, en approchant la dérivée u' par des différences finies.

- «À un pas» : on n'utilise **qu'une seule valeur passée** pour calculer u_{n+1}
- ▶ *Explicite* : u_{n+1} est directement donné par une fonction $\phi(t_n, u_n, \delta t_n)$
 - ▶ *Implicite* : u_{n+1} est une fonction $\phi(t_n, u_n, u_{n+1}, \delta t_n)$. Il faut alors résoudre une équation pour trouver u_{n+1} .

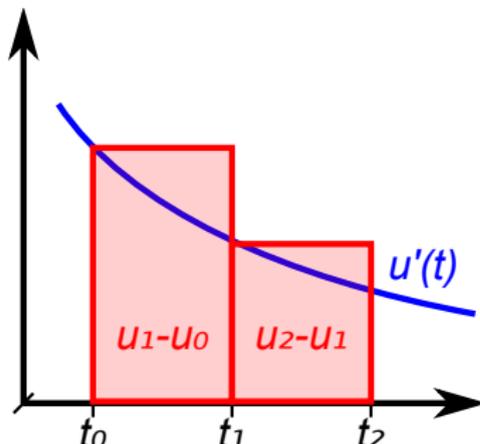
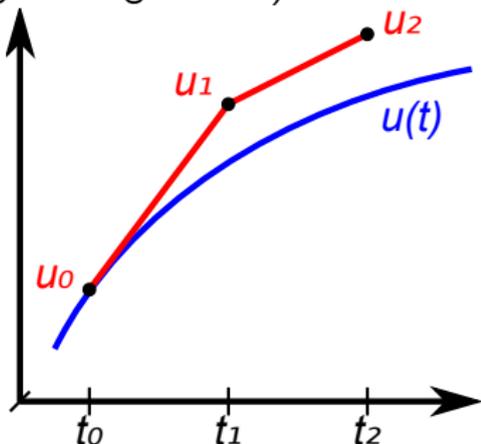
Méthode d'Euler explicite

On utilise u_n pour calculer u_{n+1} .

Le DL $u_{n+1} = u_n + \delta t u'_n + \mathcal{O}(\delta t^2)$ donne directement

$$u_{n+1} = u_n + \delta t u'_n = u_n + \delta t f(t_n, u_n) \quad (16)$$

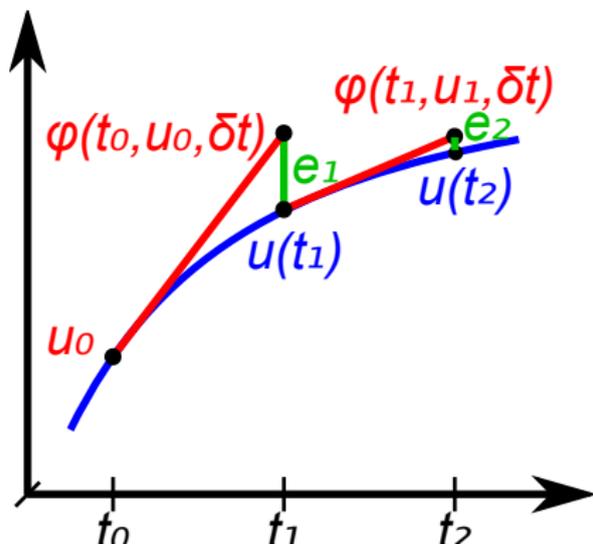
Si f ne dépend que de t , cela revient à remplacer la courbe $u(t)$ entre t_n et t_{n+1} par sa tangente en t_n (ou à intégrer $u' = f$ par la méthode des rectangles «à gauche»).



Définition des erreurs

Pour une équation de solution exacte $u(t)$ et dont la solution numérique (u_n) serait obtenue par $u_{n+1} = \phi(t_n, u_n, \delta t)$:

- ▶ *erreur locale* : après 1 pas de temps à partir de la solution exacte en t_{n-1} : $e_n = \phi(t_{n-1}, u(t_{n-1}), \delta t) - u(t_n)$;



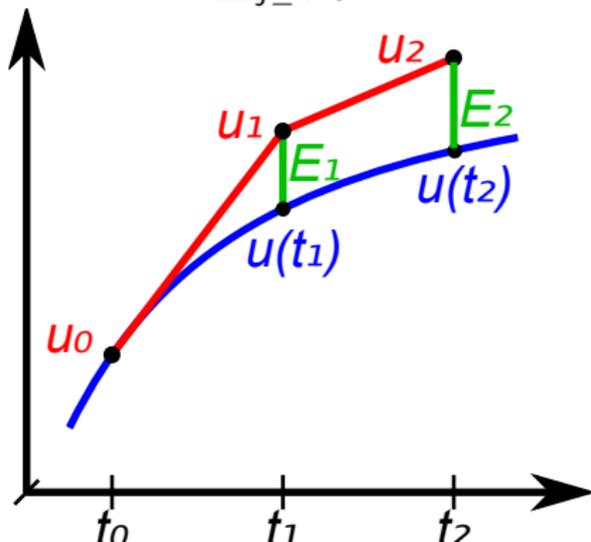
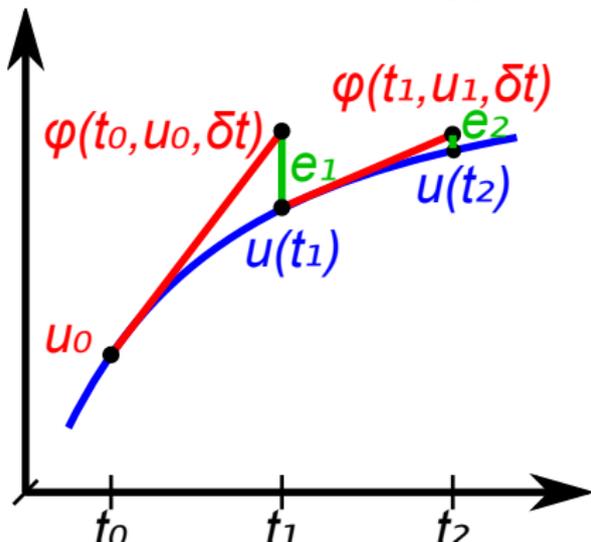
Définition des erreurs

Pour une équation de solution exacte $u(t)$ et dont la solution numérique (u_n) serait obtenue par $u_{n+1} = \phi(t_n, u_n, \delta t)$:

- ▶ *erreur locale* : après 1 pas de temps à partir de la solution exacte en t_{n-1} : $e_n = \phi(t_{n-1}, u(t_{n-1}), \delta t) - u(t_n)$;
- ▶ *erreur globale* : après n pas de temps à partir de la condition initiale : $E_n = u_n - u(t_n)$



1. il faut connaître $u(t)$. 2. $E_1 = e_1$ mais $E_n \neq \sum_{j \leq n} e_j$ (en général).



Propriétés des méthodes numériques

Une bonne méthode de résolution numérique d'équation différentielle est

- ▶ *Consistante* : quand $\delta t \rightarrow 0$, la solution *locale* de l'éq. discrétisée tend vers la solution exacte de l'équation continue plus vite que δt :

$$\forall n, \lim_{\delta t \rightarrow 0} (e_n / \delta t) = 0 \quad (17)$$

Propriétés des méthodes numériques

Une bonne méthode de résolution numérique d'équation différentielle est

- ▶ *Consistante* : quand $\delta t \rightarrow 0$, la solution *locale* de l'éq. discrétisée tend vers la solution exacte de l'équation continue plus vite que δt :

$$\forall n, \lim_{\delta t \rightarrow 0} (e_n / \delta t) = 0 \quad (17)$$

- ▶ *Convergente* : quand $\delta t \rightarrow 0$, la solution de l'équation discrétisée tend vers la solution exacte de l'équation continue :

$$\forall n, \lim_{\delta t \rightarrow 0} E_n = 0 \quad (18)$$

Propriétés des méthodes numériques

Une bonne méthode de résolution numérique d'équation différentielle est

- ▶ *Consistante* : quand $\delta t \rightarrow 0$, la solution *locale* de l'éq. discrétisée tend vers la solution exacte de l'équation continue plus vite que δt :

$$\forall n, \lim_{\delta t \rightarrow 0} (e_n / \delta t) = 0 \quad (17)$$

- ▶ *Convergente* : quand $\delta t \rightarrow 0$, la solution de l'équation discrétisée tend vers la solution exacte de l'équation continue :

$$\forall n, \lim_{\delta t \rightarrow 0} E_n = 0 \quad (18)$$

- ▶ *Stable* : pour un δt donné, la différence entre la solution numérique et la solution exacte reste bornée :

$$\exists K : \forall n, |E_n| < K \quad (19)$$

Propriétés des méthodes numériques

Une bonne méthode de résolution numérique d'équation différentielle est

- ▶ **Consistante** : quand $\delta t \rightarrow 0$, la solution *locale* de l'éq. discrétisée tend vers la solution exacte de l'équation continue plus vite que δt :

$$\forall n, \lim_{\delta t \rightarrow 0} (e_n / \delta t) = 0 \quad (17)$$

- ▶ **Convergente** : quand $\delta t \rightarrow 0$, la solution de l'équation discrétisée tend vers la solution exacte de l'équation continue :

$$\forall n, \lim_{\delta t \rightarrow 0} E_n = 0 \quad (18)$$

- ▶ **Stable** : pour un δt donné, la différence entre la solution numérique et la solution exacte reste bornée :

$$\exists K : \forall n, |E_n| < K \quad (19)$$

Pour les méthodes à un pas : si l'erreur locale est en $\mathcal{O}(\delta t^{p+1})$, l'erreur globale est en (δt^p) (la méthode est d'ordre p).

De manière générale (th. Lax) : **consistance + stabilité \Rightarrow convergence.**

Analyse de la méthode d'Euler explicite

Un DL d'ordre 1 de e_n prouve la *consistance* :

$$e_n(\delta t) = u(t_n) + \delta t f(t_n, u(t_n)) - u(t_{n+1}) \quad (20)$$

$$= (u(t_n) + \delta t u'(t_n)) - (u(t_n) + \delta t u'(t_n) + \mathcal{O}(\delta t^2)) \quad (21)$$

$$= \mathcal{O}(\delta t^2) \quad (22)$$

Analyse de la méthode d'Euler explicite

Un DL d'ordre 1 de e_n prouve la *consistance* :

$$e_n(\delta t) = u(t_n) + \delta t f(t_n, u(t_n)) - u(t_{n+1}) \quad (20)$$

$$= (u(t_n) + \delta t u'(t_n)) - (u(t_n) + \delta t u'(t_n) + \mathcal{O}(\delta t^2)) \quad (21)$$

$$= \mathcal{O}(\delta t^2) \quad (22)$$

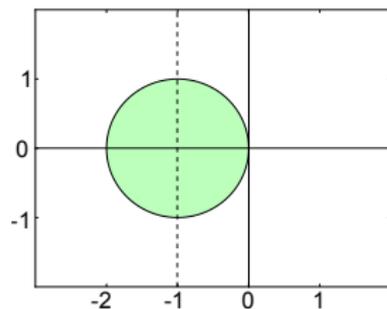
Pour la *stabilité*, on se place dans un cas particulier (linéarisation de l'équation différentielle originale) : $f(t, u) = k u$ et $u(0) = u_0$.

Alors l'étude de la stabilité revient à voir si $|u_{n+1}/u_n| < 1$.

$$u_{n+1} = u_n + k \delta t u_n = (1 + k \delta t) u_n \quad (23)$$

$$u_{n+1}/u_n = 1 + k \delta t \quad (24)$$

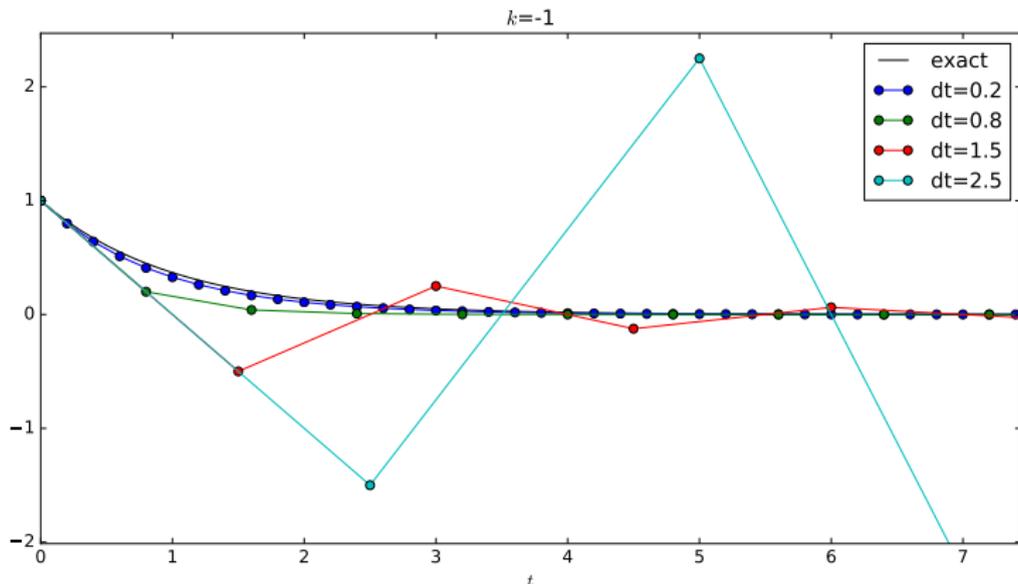
Stable ssi $|1 + k \delta t| \leq 1$.



À quoi ressemble l'instabilité ?

Pour $f(t, u) = k u$ et $u(0) = u_0$, pour $k \in \mathbb{R}$ et δt donnés :

- ▶ $k > 0$ ($k \delta t > 0$) : solution exacte exponentielle croissante : la solution numérique « n'arrive pas à suivre », E_n non borné.
- ▶ $k \delta t < -2$: solution numérique oscillante de val. abs. croissante



Méthode d'Euler implicite

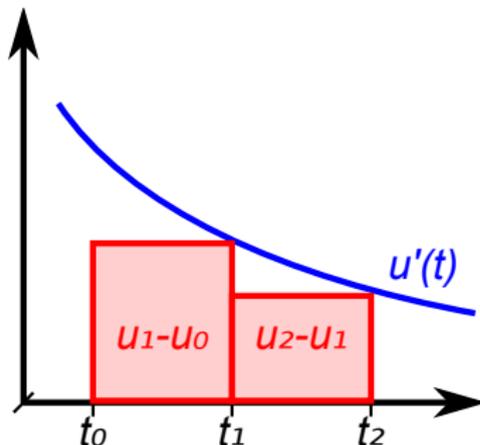
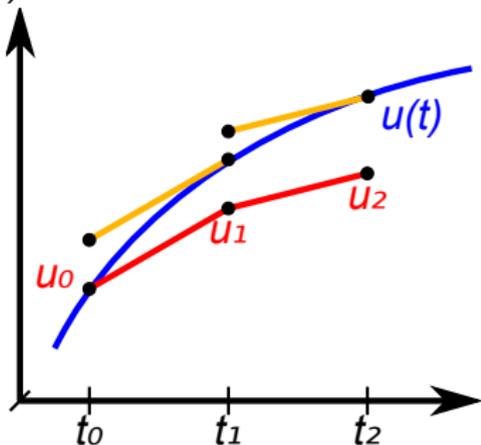
On veut encore utiliser u_n pour calculer u_{n+1} .

Mais on écrit le DL pour t_n «en arrière», en partant de t_{n+1} :

$u_n = u_{n+1} - \delta t u'_{n+1} + \mathcal{O}(\delta t^2)$. Cela donne :

$$u_{n+1} = u_n + \delta t u'_{n+1} = u_n + \delta t f(t_{n+1}, u_{n+1}) \quad (25)$$

Si f ne dépend que de t , cela revient à remplacer la courbe $u(t)$ entre t_n et t_{n+1} par un segment passant par (t_n, u_n) et parallèle à sa tangente (*estimée*) en t_{n+1} (ou à intégrer $u' = f$ par la méthode des rectangles «à droite»).



Analyse de la méthode d'Euler implicite

Dans le cas particulier $f(t, u) = k u$ et $u(0) = u_0$: $u(t) = u_0 \exp(k t)$

$$u_{n+1} = u_n + k \delta t u_{n+1} \quad (26)$$

$$u_{n+1} = u_n / (1 - k \delta t) \quad (27)$$

$$e_n = u(t_n) / (1 - k \delta t) - u(t_{n+1}) \quad (28)$$

$$= u(t_n) (1 / (1 - k \delta t) - \exp(k \delta t)) = \mathcal{O}(\delta t^2) \quad (29)$$

ce qui montre la *consistance*.

Analyse de la méthode d'Euler implicite

Dans le cas particulier $f(t, u) = k u$ et $u(0) = u_0$: $u(t) = u_0 \exp(k t)$

$$u_{n+1} = u_n + k \delta t u_{n+1} \quad (26)$$

$$u_{n+1} = u_n / (1 - k \delta t) \quad (27)$$

$$e_n = u(t_n) / (1 - k \delta t) - u(t_{n+1}) \quad (28)$$

$$= u(t_n) (1 / (1 - k \delta t) - \exp(k \delta t)) = \mathcal{O}(\delta t^2) \quad (29)$$

ce qui montre la *consistance*.

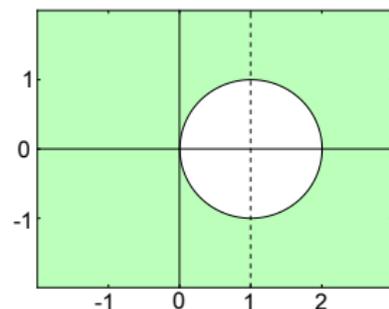
Pour la *stabilité*,

$$u_{n+1} = u_n + k \delta t u_{n+1} \quad (30)$$

$$u_{n+1} / u_n = 1 / (1 - k \delta t) \quad (31)$$

Stable ssi $|1 - k \delta t| \geq 1$:
toujours stable pour $k < 0$.

δt peut être plus long que $1/k$ (échelle de temps de l'équation) \rightarrow permet de mieux résoudre des équations "raides"



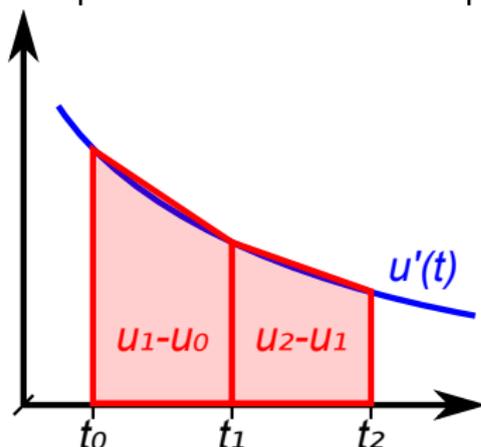
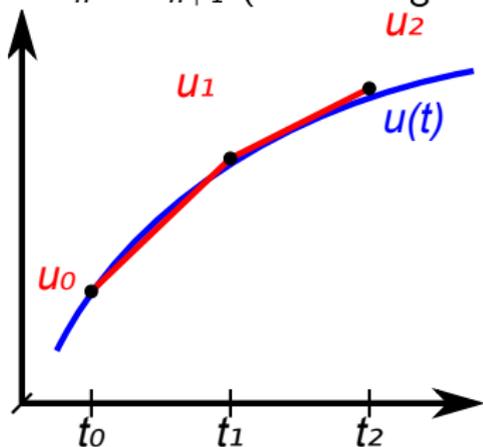
Méthode de Crank-Nicolson

On fait la demi-différence des DL d'ordre 1

$u_{n+1} = u(t_n + \delta t)$ et $u_n = u(t_{n+1} - \delta t)$, ce qui donne

$$u_{n+1} = u_n + \frac{\delta t}{2} (f(t_n, u_n) + f(t_{n+1}, u_{n+1})) \quad (32)$$

Si f ne dépend que de t , cela revient à remplacer la courbe $u(t)$ entre t_n et t_{n+1} par un segment passant par (t_n, u_n) et de pente la moyenne des pentes en t_n et t_{n+1} (ou à intégrer $u' = f$ par la méthode des trapèzes).



Analyse de la méthode de Crank-Nicolson

Dans le cas particulier $f(t, u) = k u$ et $u(0) = u_0$: $u(t) = u_0 \exp(k t)$

$$u_{n+1} = u_n \frac{1 + k \delta t/2}{1 - k \delta t/2} \quad (33)$$

$$e_n = u(t_n) \left(\frac{1 + k \delta t/2}{1 - k \delta t/2} - \exp(k \delta t) \right) = \mathcal{O}(\delta t^3) \quad (34)$$

ce qui montre la *consistance* (et l'ordre 2).

Analyse de la méthode de Crank-Nicolson

Dans le cas particulier $f(t, u) = k u$ et $u(0) = u_0$: $u(t) = u_0 \exp(k t)$

$$u_{n+1} = u_n \frac{1 + k \delta t/2}{1 - k \delta t/2} \quad (33)$$

$$e_n = u(t_n) \left(\frac{1 + k \delta t/2}{1 - k \delta t/2} - \exp(k \delta t) \right) = \mathcal{O}(\delta t^3) \quad (34)$$

ce qui montre la *consistance* (et l'ordre 2).

Le schéma est *stable* ssi

$$\left| \frac{1 + k \delta t/2}{1 - k \delta t/2} \right| \leq 1 \quad (35)$$

$$|k \delta t/2 - (-1)| \leq |k \delta t/2 - (+1)| \quad (36)$$

$$\Re(k) \leq 0 \quad (37)$$

Méthodes à plusieurs pas

On utilise **plusieurs valeurs passées** u_i et $u'_i = f(t_i, u_i)$, $i \in [n+1-s, n]$.

► Explicite : méthodes d'Adams-Bashforth, par exemple

$$u_{n+1} = u_n + \delta t \left(\frac{3}{2} u'_n - \frac{1}{2} u'_{n-1} \right) \quad (s=2) \quad (38)$$

$$u_{n+1} = u_n + \delta t \left(\frac{23}{12} u'_n - \frac{4}{3} u'_{n-1} + \frac{5}{12} u'_{n-2} \right) \quad (s=3) \quad (39)$$

obtenues par DL ou par approximation polynomiale en t_{n+1-s}, \dots, t_n
de l'intégrande de

$$u_{n+1} = u_n + \int_{t_n}^{t_{n+1}} f(t, u(t)) dt \quad (40)$$

Méthodes à plusieurs pas

On utilise **plusieurs valeurs passées** u_i et $u'_i = f(t_i, u_i)$, $i \in [n+1-s, n]$.

- Explicite : méthodes d'Adams-Bashforth, par exemple

$$u_{n+1} = u_n + \delta t \left(\frac{3}{2} u'_n - \frac{1}{2} u'_{n-1} \right) \quad (s=2) \quad (38)$$

$$u_{n+1} = u_n + \delta t \left(\frac{23}{12} u'_n - \frac{4}{3} u'_{n-1} + \frac{5}{12} u'_{n-2} \right) \quad (s=3) \quad (39)$$

obtenues par DL ou par approximation polynomiale en t_{n+1-s}, \dots, t_n de l'intégrande de

$$u_{n+1} = u_n + \int_{t_n}^{t_{n+1}} f(t, u(t)) dt \quad (40)$$

- Implicite : méthodes d'Adams-Moulton, par exemple

$$u_{n+1} = u_n + \frac{1}{2} \delta t (u'_{n+1} + u'_n) \quad (= \text{Crank-Nicolson}) \quad (41)$$

$$u_{n+1} = u_n + \delta t \left(\frac{5}{12} u'_{n+1} + \frac{2}{3} u'_n - \frac{1}{12} u'_{n-1} \right) \quad (42)$$

Méthodes à plusieurs pas

Besoin de moins de calculs que Runge-Kutta du même ordre, mais

- ▶ Besoin de “lancer” le calcul par des méthodes utilisant moins de valeurs passées.
- ▶ Besoin de garder des valeurs passées en mémoire (lourd pour PDE).
- ▶ Méthodes (coefficients) plus difficiles à obtenir pour un pas de temps variable.

Méthodes avec pas intermédiaires : Runge-Kutta

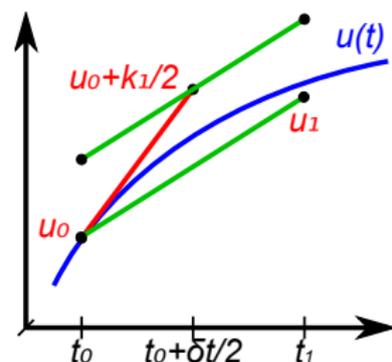
q points intermédiaires entre t_n et t_{n+1} : $t_{n,i} = t_n + c_i \delta t, i \in [1, q]$.

Par exemple avec un point intermédiaire :

$$k_1 = \delta t f(t_n, u_n) \quad (43)$$

$$k_2 = \delta t f\left(t_n + \frac{1}{2}\delta t, u_n + \frac{1}{2}k_1\right) \quad (44)$$

$$u_{n+1} = u_n + k_2 \quad (45)$$



C'est une méthode «RK2», d'ordre 2.

Une méthode «RK4», d'ordre 4, couramment utilisée est

$$k_1 = \delta t f(t_n, u_n) \quad (46)$$

$$k_2 = \delta t f\left(t_n + \delta t/2, u_n + k_1/2\right) \quad (47)$$

$$k_3 = \delta t f\left(t_n + \delta t/2, u_n + k_2/2\right) \quad (48)$$

$$k_4 = \delta t f\left(t_n + \delta t, u_n + k_3\right) \quad (49)$$

$$u_{n+1} = u_n + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4) \quad (50)$$

Méthodes «prédicteur-correcteur»

- ▶ *Prédicteur* : première évaluation $u_{n+1}^{(0)}$ de u_{n+1} par une méthode explicite.
- ▶ *Correcteur* : itération d'une méthode implicite.
Par exemple si on part de Crank-Nicolson :

$$u_{n+1}^{(k+1)} = u_n + \frac{\delta t}{2} \left(f(t_n, u_n) + f(t_{n+1}, u_{n+1}^{(k)}) \right) \quad (51)$$

Comme $u_{n+1}^{(k)}$ est déjà connu lors du calcul de $u_{n+1}^{(k+1)}$, on n'a plus la difficulté habituelle des méthodes implicites.

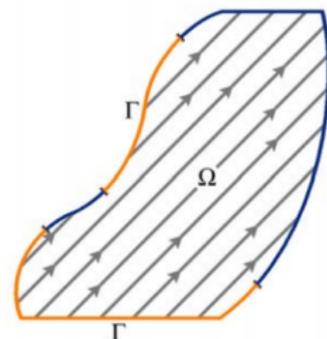
Si l'ordre du correcteur est n et celle du prédicteur est au moins $n - 1$, l'ordre du prédicteur-correcteur sera n

Équations aux dérivées partielles

1. *Approximation de dérivées par des différences finies*
2. *Équations différentielles ordinaires avec condition initiale*
3. *Équations aux dérivées partielles*
4. *Exemple : un tube de flux turbulent*

Résolution d'EDP par différences finies

Expressions faisant intervenir des *dérivées partielles* d'une fonction u , avec des conditions sur une partie Γ du bord $\partial\Omega$ du domaine de définition Ω de u .



MIT OCW18.336

Les dérivées peuvent être obtenues par combinaisons de valeurs en un *nombre fini de nœuds* du maillage.

- + Simple à mettre en œuvre, faible coût en temps de calcul, bons résultats théoriques sur la convergence
- Difficile de prendre en compte des géométries complexes ou certaines conditions aux limites, et non conservatif.

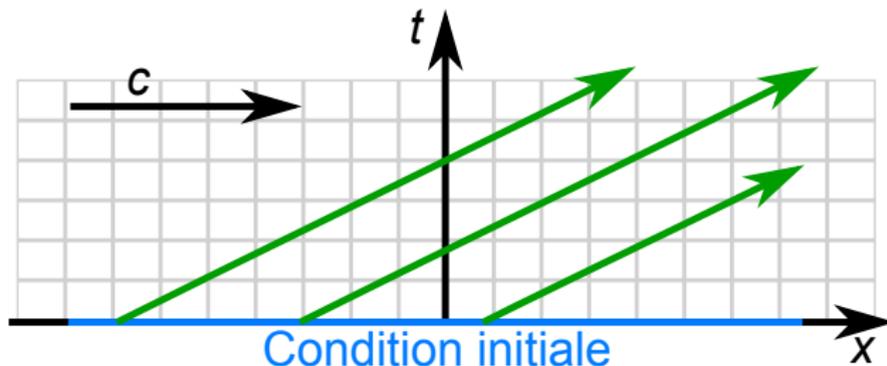
Exemple d'équation à résoudre

Advection linéaire (propagation d'une onde d'Alfvén...) :

$$\partial_t u + c \partial_x u = 0, \text{ avec la condition initiale } u(x, 0) = u_0(x) \quad (52)$$

de solution exacte

$$u(x, t) = u_0(x - ct) \quad (53)$$



Discrétisation : $u_j^n = u(j \delta x, n \delta t)$.

Nombre de Courant : $r = c \delta t / \delta x$.

Schémas explicites les plus simples

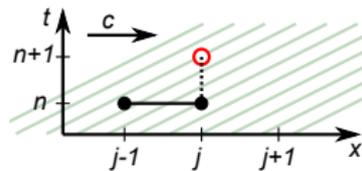
Le problème : on connaît les u_j^n , comment calculer les u_j^{n+1} ?

Méthodes explicites au premier ordre en temps :

$\partial_t u_j^n$ approché par $(u_j^{n+1} - u_j^n)/\delta t + \mathcal{O}(\delta t)$, et $\partial_x u_j^n$ par :

► « *Upwind* » : $(u_j^n - u_{j-1}^n)/\delta x + \mathcal{O}(\delta x)$

d'où $(u_j^{n+1})_{\text{num}} = (1 - r)u_j^n + ru_{j-1}^n$



Schémas explicites les plus simples

Le problème : on connaît les u_j^n , comment calculer les u_j^{n+1} ?

Méthodes explicites au premier ordre en temps :

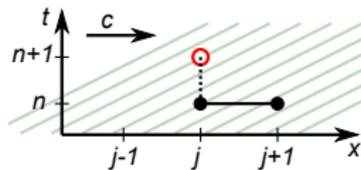
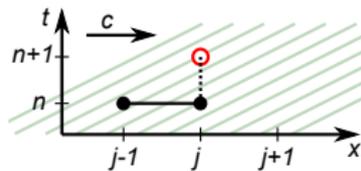
$\partial_t u_j^n$ approché par $(u_j^{n+1} - u_j^n)/\delta t + \mathcal{O}(\delta t)$, et $\partial_x u_j^n$ par :

► « *Upwind* » : $(u_j^n - u_{j-1}^n)/\delta x + \mathcal{O}(\delta x)$

$$\text{d'où } (u_j^{n+1})_{\text{num}} = (1 - r)u_j^n + ru_{j-1}^n$$

► « *Downwind* » : $(u_{j+1}^n - u_j^n)/\delta x + \mathcal{O}(\delta x)$

$$\text{d'où } (u_j^{n+1})_{\text{num}} = (1 + r)u_j^n - ru_{j+1}^n$$



Schémas explicites les plus simples

Le problème : on connaît les u_j^n , comment calculer les u_j^{n+1} ?

Méthodes explicites au premier ordre en temps :

$\partial_t u_j^n$ approché par $(u_j^{n+1} - u_j^n)/\delta t + \mathcal{O}(\delta t)$, et $\partial_x u_j^n$ par :

► « *Upwind* » : $(u_j^n - u_{j-1}^n)/\delta x + \mathcal{O}(\delta x)$

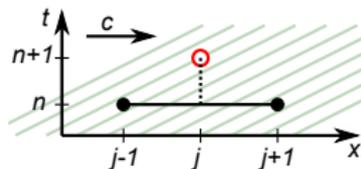
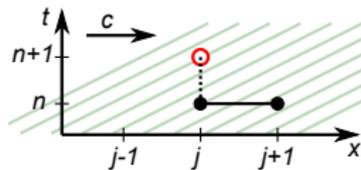
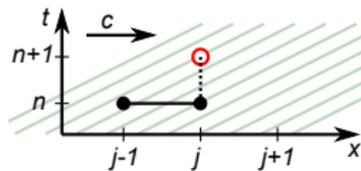
d'où $(u_j^{n+1})_{\text{num}} = (1 - r)u_j^n + ru_{j-1}^n$

► « *Downwind* » : $(u_{j+1}^n - u_j^n)/\delta x + \mathcal{O}(\delta x)$

d'où $(u_j^{n+1})_{\text{num}} = (1 + r)u_j^n - ru_{j+1}^n$

► *Centré* : $(u_{j+1}^n - u_{j-1}^n)/2\delta x + \mathcal{O}(\delta x^2)$

d'où $(u_j^{n+1})_{\text{num}} = u_j^n - \frac{r}{2}(u_{j+1}^n - u_{j-1}^n)$



Analyse de stabilité

Étude de l'amplitude ξ^n d'une onde : $u_j^n = \xi^n \exp(i k j \delta x)$:
amplification $\xi = \xi^{n+1}/\xi^n$ (devrait être égale à $\exp(-i k c \delta t)$).

- ▶ «Upwind» :

$$(u_j^{n+1})_{\text{num}} = \xi^n e^{i k j \delta x} (1 - r + r e^{-i k \delta x}) = \xi^{n+1} e^{i k j \delta x} \quad (54)$$

$$\xi^{n+1}/\xi^n = 1 - r + r \exp(-i k \delta x) \quad (55)$$

$$|\xi| \leq 1 \quad \forall k \quad (\text{stable pour } r \in [0, 1]) \quad (56)$$

- ▶ «Downwind» :

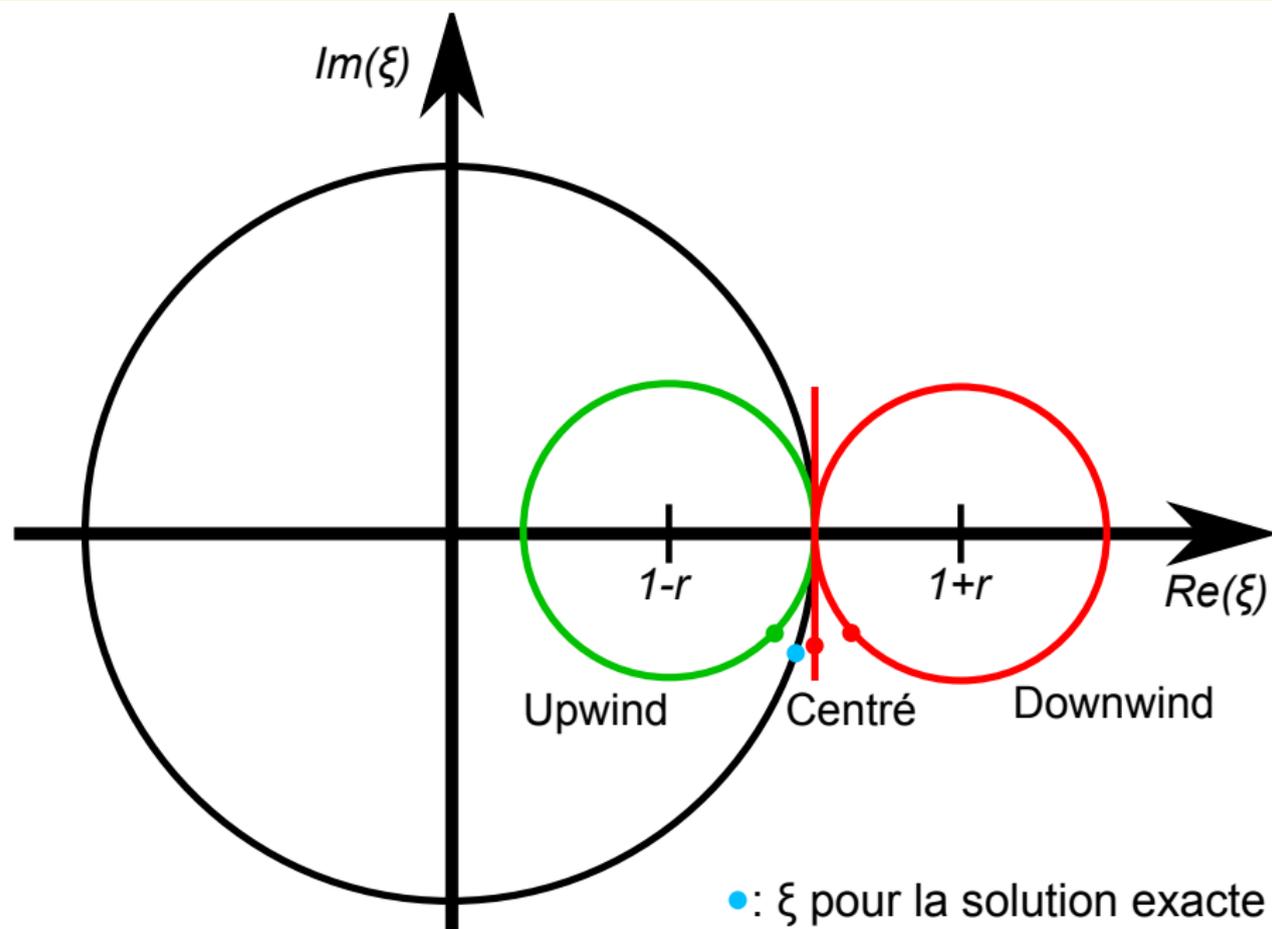
$$\xi = 1 + r - r \exp(i k \delta x) \quad (57)$$

$$\forall r > 0, \exists k : |\xi| > 1 \quad (\text{inconditionnellement } \textit{instable}) \quad (58)$$

- ▶ Centré :

$$\xi = 1 - i r \sin k \delta x \quad (\text{inconditionnellement } \textit{instable}) \quad (59)$$

Analyse de stabilité



Que faire pour rendre le schéma centré stable ?

Le DL en δt de u_j^{n+1} est :

$$u_j^{n+1} = u_j^n + \delta t \partial_t u_j^n + \frac{\delta t^2}{2} \partial_t^2 u_j^n + \mathcal{O}(\delta t^3) \quad (60)$$

$$= u_j^n - r \delta x \partial_x u_j^n + \frac{r^2}{2} \delta x^2 \partial_x^2 u_j^n + \mathcal{O}(\delta x^3) \quad (61)$$

En faisant un DL en δx de la solution numérique $u_j^n - \frac{r}{2}(u_{j+1}^n - u_{j-1}^n)$:

$$(u_j^{n+1})_{\text{num}} = u_j^n - r \delta x \partial_x u_j^n + \mathcal{O}(\delta x^3) \quad (62)$$

L'erreur locale est alors

$$e_n = \frac{r^2}{2} \delta x^2 \partial_x^2 u_j^n + \mathcal{O}(\delta x^3) \quad (63)$$

Que faire pour rendre le schéma centré stable ?

Le DL en δt de u_j^{n+1} est :

$$u_j^{n+1} = u_j^n + \delta t \partial_t u_j^n + \frac{\delta t^2}{2} \partial_t^2 u_j^n + \mathcal{O}(\delta t^3) \quad (60)$$

$$= u_j^n - r \delta x \partial_x u_j^n + \frac{r^2}{2} \delta x^2 \partial_x^2 u_j^n + \mathcal{O}(\delta x^3) \quad (61)$$

En faisant un DL en δx de la solution numérique $u_j^n - \frac{r}{2}(u_{j+1}^n - u_{j-1}^n)$:

$$(u_j^{n+1})_{\text{num}} = u_j^n - r \delta x \partial_x u_j^n + \mathcal{O}(\delta x^3) \quad (62)$$

L'erreur locale est alors

$$e_n = \frac{r^2}{2} \delta x^2 \partial_x^2 u_j^n + \mathcal{O}(\delta x^3) \quad (63)$$

Cela correspond à un *terme de diffusion*, de coefficient $-\frac{r^2}{2} \delta x^2 \delta t$.

Idée : ajouter une telle diffusion ; donne le schéma de *Lax-Wendroff* :

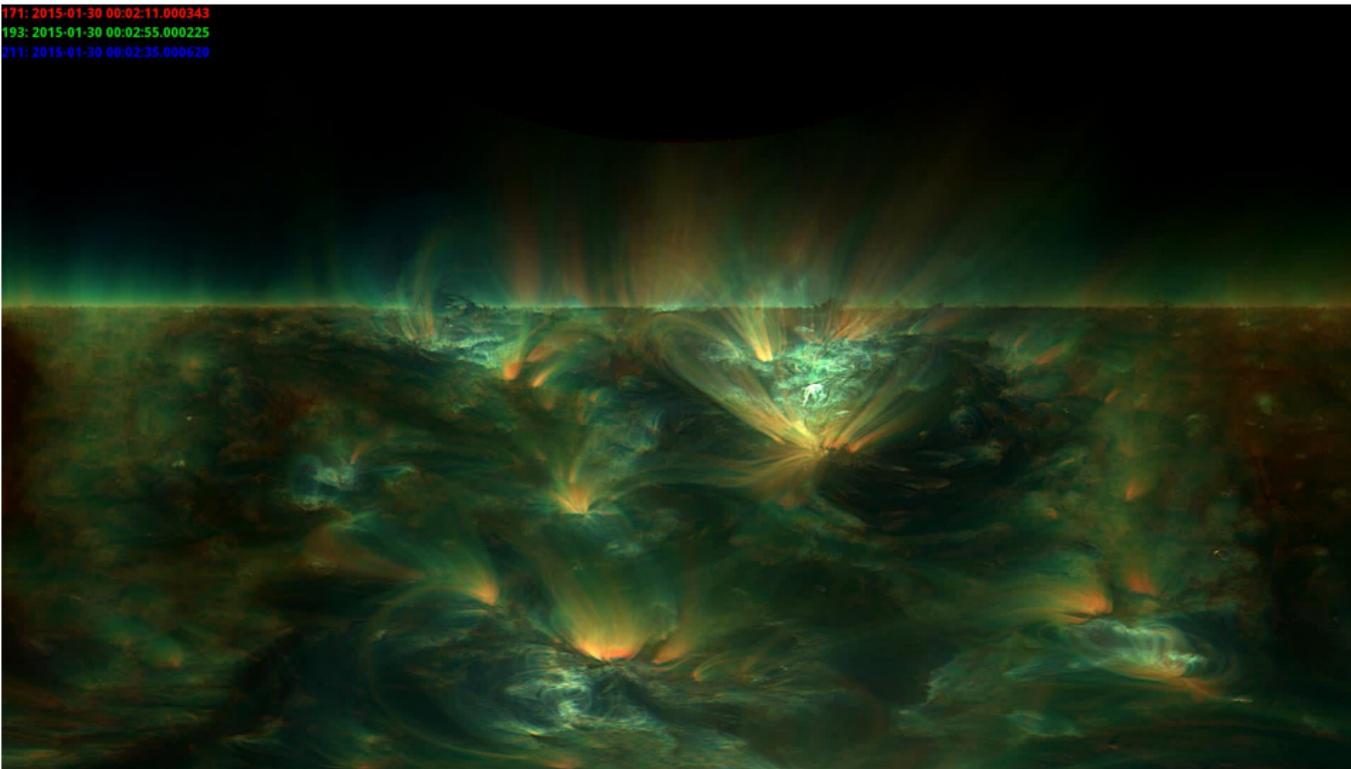
$$(u_j^{n+1})_{\text{num}} = u_j^n - \frac{r}{2}(u_{j+1}^n - u_{j-1}^n) + \frac{r^2}{2}(u_{j-1}^n - 2u_j^n + u_{j+1}^n) \quad (64)$$

Exemple : un tube de flux turbulent

- 1. Approximation de dérivées par des différences finies*
- 2. Équations différentielles ordinaires avec condition initiale*
- 3. Équations aux dérivées partielles*
- 4. Exemple : un tube de flux turbulent*

Motivation : activité et chauffage de la couronne solaire

171: 2015-01-30 00:02:11.000343
193: 2015-01-30 00:02:55.000225
211: 2015-01-30 00:02:35.000620

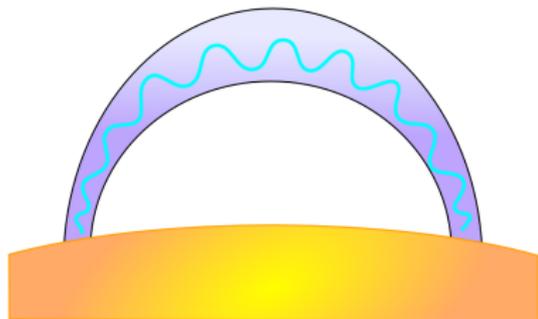


SDO/AIA

Un tube de flux : empilement de modèles en couches

(couplés par des ondes d'Alfvén)

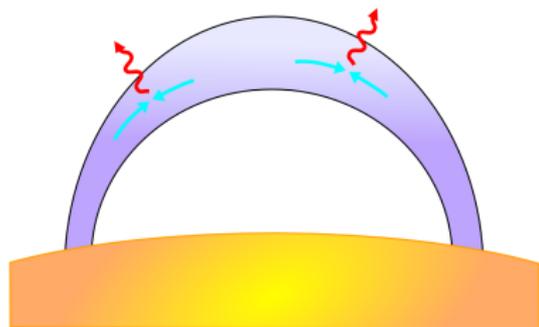
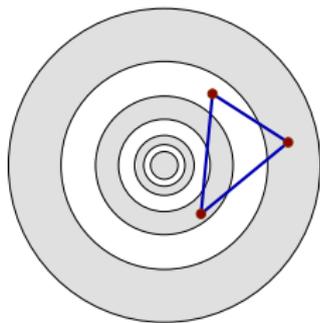
- ▶ Géométrie imposée
(ici une boucle coronale)
- ▶ Conditions aux limites
(ici champ de vitesse)
- ▶ Ondes d'Alfvén



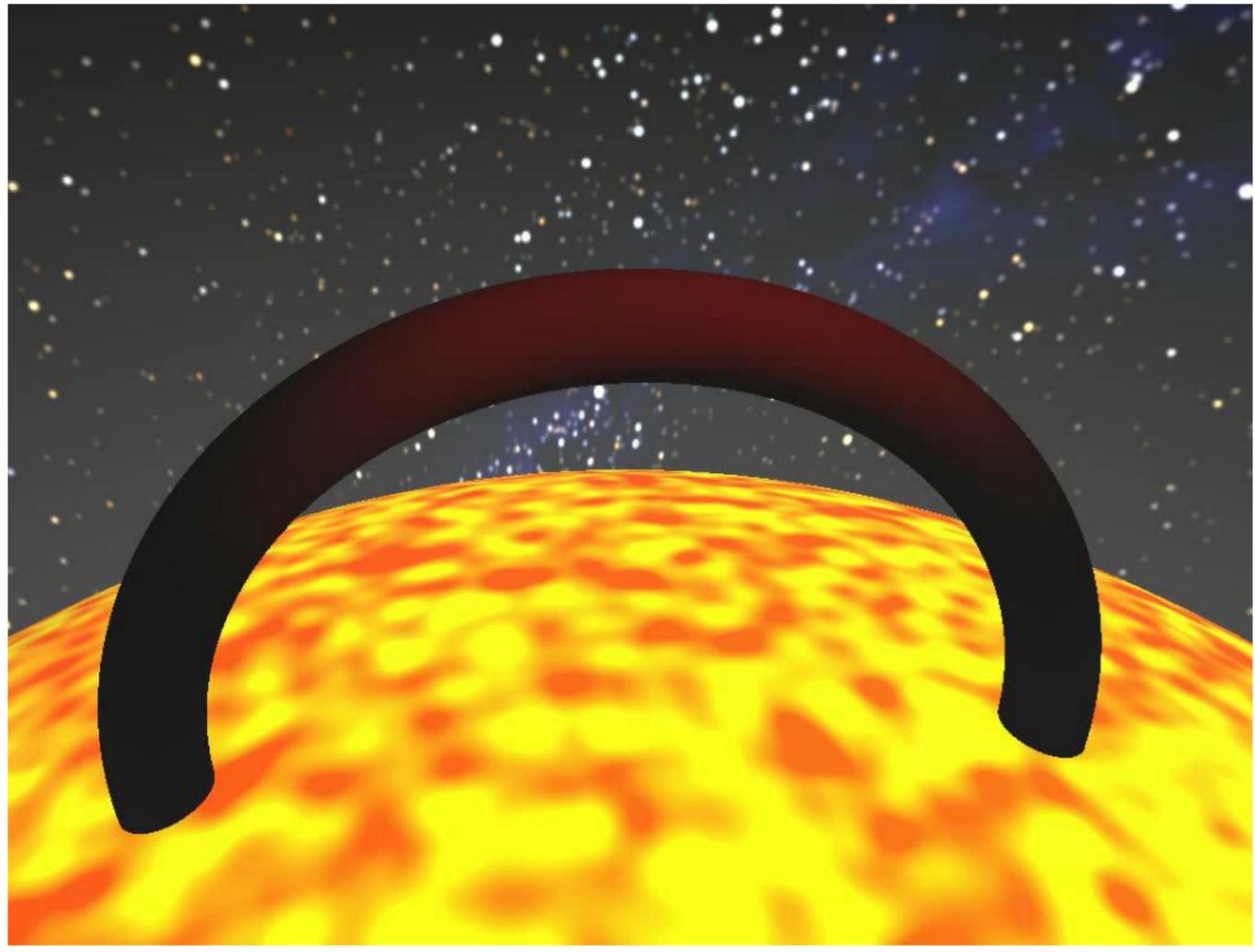
Un tube de flux : empilement de modèles en couches

(couplés par des ondes d'Alfvén)

- ▶ Géométrie imposée
(ici une boucle coronale)
- ▶ Conditions aux limites
(ici champ de vitesse)
- ▶ Ondes d'Alfvén
- ▶ Termes non-linéaires de la MHD,
turbulence : « Modèles en couches »



Avantage : modèle simplifié de turbulence à *grand Reynolds* ($\approx 10^6$)



- ▶ Essentiellement *Fortran 90*
- ▶ *Différences finies*
- ▶ *Modulaire*
(différents types de conditions physiques, de schémas numériques...)

- ▶ Essentiellement *Fortran 90*
- ▶ *Différences finies*
- ▶ *Modulaire*
(différents types de conditions physiques, de schémas numériques...)

Besoin de parallélisation :

La turbulence est certes modélisée de façon simplifiée, mais $\tau_A/\tau_\nu \gg 1$, donc besoin d'une grande résolution le long du tube de flux :

des milliers de « modèles en couches » !

Le mini-projet

- ▶ Prendre en main le code sans parallélisation, le faire tourner sur un processeur
- ▶ Le paralléliser, faire des courbes de scaling
- ▶ Faire tourner le code sur beaucoup de processeurs
- ▶ Visualiser les résultats
- ▶ Films de la dynamique de la cascade turbulente.

Le mini-projet

- ▶ Prendre en main le code sans parallélisation, le faire tourner sur un processeur
- ▶ Le paralléliser, faire des courbes de scaling
- ▶ Faire tourner le code sur beaucoup de processeurs
- ▶ Visualiser les résultats
- ▶ Films de la dynamique de la cascade turbulente.

À vous de jouer !

Ces transparents :

<http://eric.buchlin.org/teaching/index.fr#simulations>

Ce qui est aussi inclus

Aussi inclus :

stratification (imposée) de $\rho / v_A / B_0 /$ expansion du tube de flux

Ce qui est aussi inclus ... et ce qui ne l'est pas

Aussi inclus :

stratification (imposée) de ρ / v_A / B_0 / expansion du tube de flux

Pas inclus (mais possible par couplage avec d'autres codes) :

- ▶ Flots longitudinaux, fluctuations de B_0
- ▶ Compressibilité
- ▶ Effets non-MHD : fluides multiples, particules non-thermiques
- ▶ Gravité (mais stratification de ρ)
- ▶ Thermodynamique

Alternatives de modules

(choisies par des variables dans un fichier inclus dans le Makefile) :

- ▶ Type de stratification
- ▶ Conditions aux limites
- ▶ Formats d'entrées/sorties
- ▶ Schémas numériques pour la propagation des ondes
- ▶ Modèle pour les non-linéarités (pas forcément modèles en couches!)
- ▶ (Dés-)Activation du parallélisme
(pas besoin de librairie MPI si 1 CPU)

Adaptation semi-automatique à la plate-forme
(système, compilateur, parallélisme)

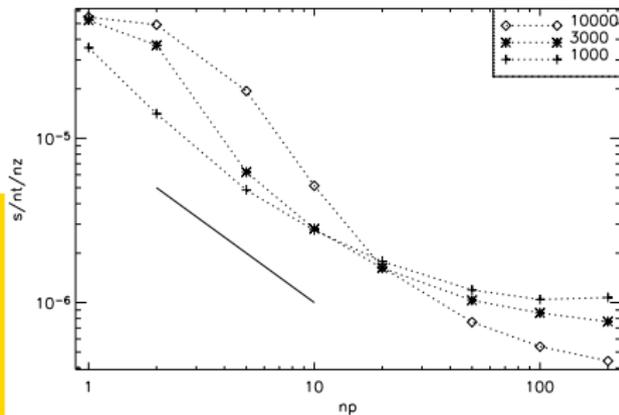
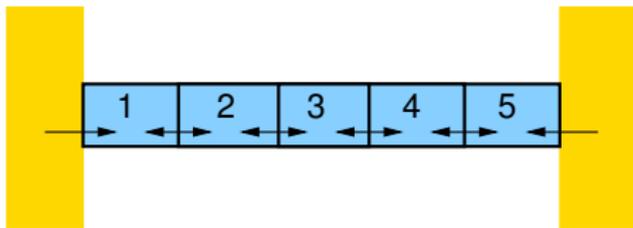
Parallélisme

Large intervalle d'échelles de temps :

$1/k_{\parallel} v_A$ (Alfvén); $1/k_{\perp} (u_{\perp} \pm b_{\perp})$ (non-linéaire); $1/\nu k_{\perp}^2$ (dissipation)

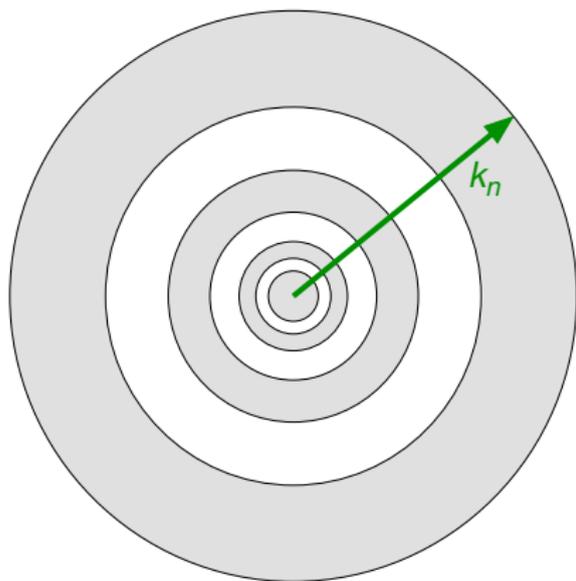
→ aussi besoin d'un assez grand nombre de plans pour la propagation des ondes, → plus lourd que ça en a l'air...

Typiquement, pour $N_{\perp} = 20$, on doit avoir $N_{\parallel} = 1000$



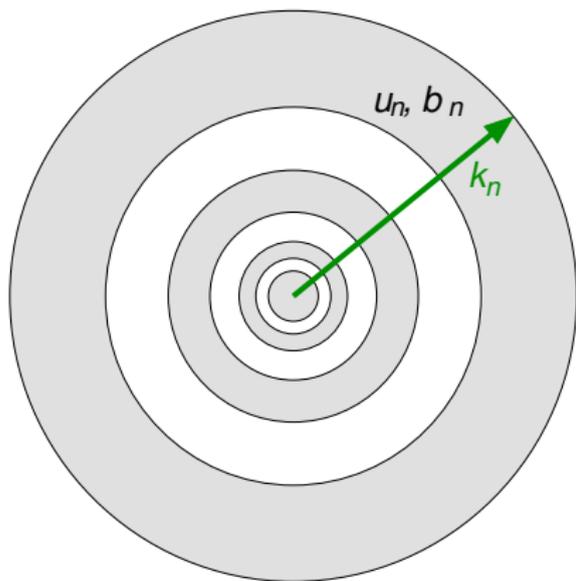
Construction d'un modèle en couches MHD

- ▶ *Couches concentriques* dans l'espace de Fourier : $k_n = k_0 \lambda^n$



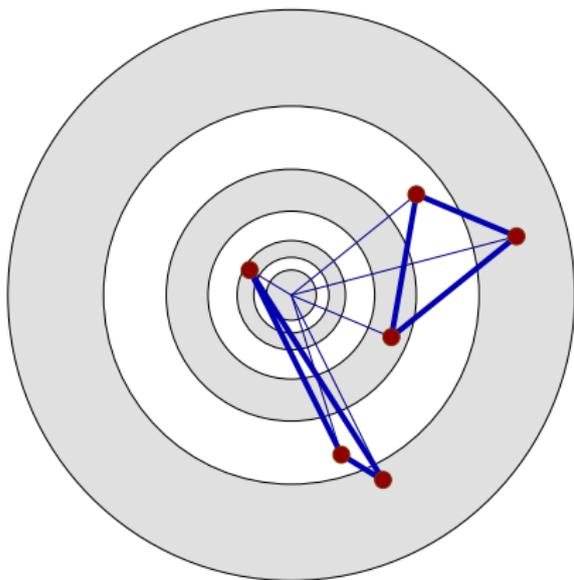
Construction d'un modèle en couches MHD

- ▶ *Couches concentriques* dans l'espace de Fourier : $k_n = k_0 \lambda^n$
- ▶ Variables $(u_n, b_n)_n \in \mathbb{C}^{2N}$ pour les champs



Construction d'un modèle en couches MHD

- ▶ *Couches concentriques* dans l'espace de Fourier : $k_n = k_0 \lambda^n$
- ▶ Variables $(u_n, b_n)_n \in \mathbb{C}^{2N}$ pour les champs
- ▶ Termes non-linéaires de la MHD \rightarrow *interactions* entre 3 modes

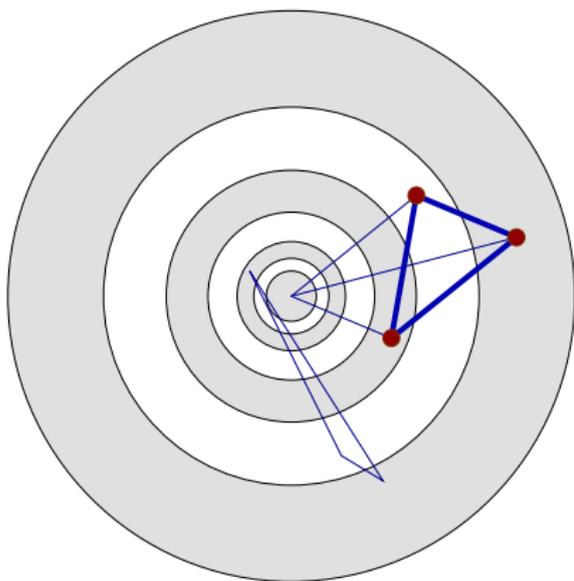


Avec $Z_n^\pm = u_n \pm b_n$:

$$d_t Z_n^\pm = ik_n \left(\sum_{l,m} \alpha_{l,m} Z_l^\pm Z_m^\mp \right)^*$$

Construction d'un modèle en couches MHD

- ▶ *Couches concentriques* dans l'espace de Fourier : $k_n = k_0 \lambda^n$
- ▶ Variables $(u_n, b_n)_n \in \mathbb{C}^{2N}$ pour les champs
- ▶ Termes non-linéaires de la MHD \rightarrow *interactions* entre 3 modes
- ▶ Hypothèse : interactions *locales*



Avec $Z_n^\pm = u_n \pm b_n$:

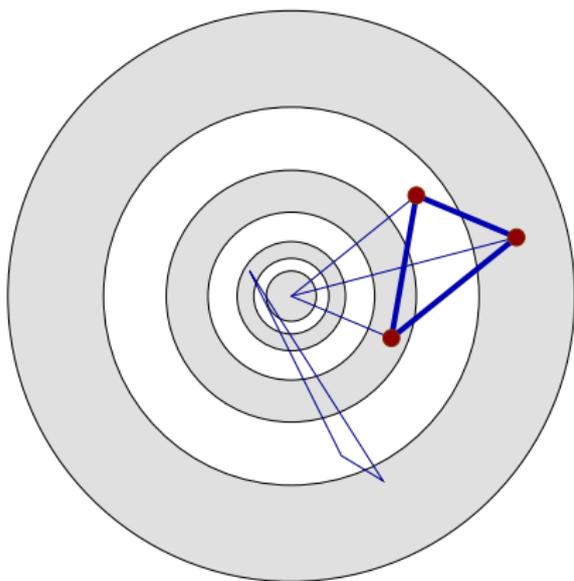
$$d_t Z_n^\pm = ik_n \left(\sum_{l,m} \alpha_{l,m} Z_l^\pm Z_m^\mp \right)^*$$

Somme restreinte à

$$\{l, m\} \in \{ \{n-2, n-1\}, \{n-1, n+1\}, \{n+1, n+2\} \}$$

Construction d'un modèle en couches MHD

- ▶ *Couches concentriques* dans l'espace de Fourier : $k_n = k_0 \lambda^n$
- ▶ Variables $(u_n, b_n)_n \in \mathbb{C}^{2N}$ pour les champs
- ▶ Termes non-linéaires de la MHD \rightarrow *interactions* entre 3 modes
- ▶ Hypothèse : interactions *locales*
- ▶ Conservation des invariants MHD \rightarrow coefficients $\alpha_{l,m}$



$$d_t Z_n^\pm = ik_n \left(\frac{\delta + \delta_m}{2} Z_{n+1}^\pm Z_{n+2}^\mp + \frac{2 - \delta - \delta_m}{2} Z_{n+1}^\mp Z_{n+2}^\pm + \frac{\delta_m - \delta}{2\lambda} Z_{n+1}^\pm Z_{n-1}^\mp - \frac{\delta + \delta_m}{2\lambda} Z_{n+1}^\mp Z_{n-1}^\pm - \frac{\delta_m - \delta}{2\lambda^2} Z_{n-1}^\pm Z_{n-2}^\mp - \frac{2 - \delta - \delta_m}{2\lambda^2} Z_{n-1}^\mp Z_{n-2}^\pm \right)^*$$

