Kick-off projet ANR

Fφst

Sylvie Boldo – Équipe-projet ProVal

28 avril 2009

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche SACLAY - ÎLE-DE-FRANCE

Kick-off $\mathcal{F}\oint st$

- tour de table
- présentation et résultats de CerPAN
- présentation de $\mathcal{F}\oint st$

Plan

CerPAN

- projet ANR Blanc 2005–2008
- **M. Mayero**, S. Boldo, J.-C. Filliâtre, F. Clément, D. Delahaye
- Certification de Programmes d'Analyse Numérique

CerPAN

Étude d'un programme de gradient $\Rightarrow$ trop dur

CerPAN

Étude d'un programme de gradient ⇒ trop dur

Étude d'un programme de résolution d'EDP (équation des ondes)
⇒ erreur de méthode
⇒ erreur d'arrondi

CerPAN – résultats

- ajouter les flottants à Caduceus
- prouver des programmes simples
- prouver l'erreur d'arrondi de la résolution d'EDP
- des morceaux de formalisation de l'erreur de méthode de la résolution d'EDP
- tactique gappa pour automatiser les preuves flottantes
- ajouter les flottants à Frama-C

CerPAN – publications

- ARITH (2007)
- RNC (2008)
- IEEE-TC (2009)
- ICALP (2009)
- Calculemus (2009)

Plan

$\mathcal{F}\oint st$

- projet ANR Blanc 2008–2011
- **S. Boldo**, J.-C. Filliâtre, F. Clément, M. Mayero
- Formal prOofs of Scientific compuTation programs

$\mathcal{F}\oint st$ – finances

Un total de 116 480 € composés de :

	INRIA - S	INRIA - R	LIPN
Missions	21 000	6 000	14 000
Consommables	21 000	12 000	18 000
Stagiaires	8 000	4 000	8 000
Frais de gestion	2 000	880	1 600
Total	52 000	22 880	41 600

En fait 502 999 € avec les salaires.

$\mathcal{F}\oint st$ – finances

Un total de 116 480 € composés de :

	INRIA - S	INRIA - R	LIPN
Missions	21 000	6 000	14 000
Consommables	21 000	12 000	18 000
Stagiaires	8 000	4 000	8 000
Frais de gestion	2 000	880	1 600
Total	52 000	22 880	41 600

En fait 502 999 € avec les salaires.

Plus 3 640 € du pôle de compétitivité System@tic Paris Région (brevets, organisation de conférences, invitation d'experts, relations internationales. . .)

$\mathcal{F}\oint st$ – objectifs

- terminer la preuve du programme de résolution d'équation des ondes
- GÉNÉRALISER
  - autres discrétisations
  - autres EDP ou ODE
  - plus de dimensions
  - automatisations
- le retour du gradient
- une boîte à outil de preuves formelles de programmes d'analyse numérique

$\mathcal{F}\oint st$ – résultats

(Ceux communs avec CerPAN)

- prouver l'erreur d'arrondi de la résolution de l'équation des ondes
- tactique gappa pour automatiser les preuves flottantes

$\mathcal{F}\oint st$ – publications

- ICALP (2009)
- Calculemus (2009)

$\mathcal{F}\oint st$ – délivrables

- programmes numériques et leurs preuves formelles
- rapport technique
- publications
- boîte à outil et son manuel (fin 2011)

CerPAN

$\mathcal{F}\phi st$

Un peu de science

L'équation des ondes

Je cherche u de $\mathbb{R}^2$ dans $\mathbb{R}$ solution de l'équation différentielle suivante, connaissant la valeur de u et de sa dérivée pour $t = 0$:

$$\frac{\partial^2 u(x,t)}{\partial t^2} - c^2 \frac{\partial^2 u(x,t)}{\partial x^2} = 0.$$

La corde discrétisée

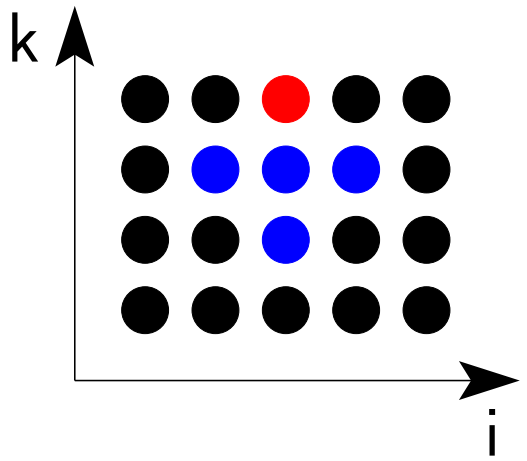
```
[...]  // initialisations de p[i][0] et p[i][1]

for (k=1; k<nk; k++) {
  p[0][k+1] = 0.;
  for (i=1; i<ni; i++) {
    dp = p[i+1][k] - 2.*p[i][k] + p[i-1][k];
    p[i][k+1] = 2.*p[i][k] - p[i][k-1] + a*dp;
   }
  p[ni][k+1] = 0.;
}
```

La corde discrétisée

```
[...]  // initialisations de p[i][0] et p[i][1]

for (k=1; k<nk; k++) {
  p[0][k+1] = 0.;
  for (i=1; i<ni; i++) {
    dp = p[i+1][k] - 2.*p[i][k] + p[i-1][k];
    p[i][k+1] = 2.*p[i][k] - p[i][k-1] + a*dp;
   }
  p[ni][k+1] = 0.;
}
```



Erreur d'arrondi

Erreur d'arrondi

Si on utilise la méthode naïve pour borner les erreurs d'arrondis, on obtient

Erreur d'arrondi

Si on utilise la méthode naïve pour borner les erreurs d'arrondis, on obtient

$$|p_i^k - exact(p_i^k)| \leq O\left(2^k 2^{-53}\right)$$

Erreur d'arrondi

Si on utilise la méthode naïve pour borner les erreurs d'arrondis, on obtient

$$|p_i^k - exact(p_i^k)| \leq O\left(2^k 2^{-53}\right)$$

C'est beaucoup (trop) car les erreurs se compensent.

Pyramide

On voit que p_i^k dépend de p_{i-1}^{k-1}, p_i^{k-1}, p_{i+1}^{k-1} et p_i^{k-2}.

Pyramide

On voit que p_i^k dépend de p_{i-1}^{k-1}, p_i^{k-1}, p_{i+1}^{k-1} et p_i^{k-2}.

Donc p_i^k dépend de la pyramide des valeurs :

$$
\begin{array}{ccccccccc}
 & & & & p_i^k & & & & \\
 & & & p_{i-1}^{k-1} & p_i^{k-1} & p_{i+1}^{k-1} & & & \\
 & & p_{i-2}^{k-2} & p_{i-1}^{k-2} & p_i^{k-2} & p_{i+1}^{k-2} & p_{i+2}^{k-2} & & \\
 & \iddots & & & \vdots & & & \ddots & \\
p_{i-k}^0 & & \cdots & & p_i^0 & & \cdots & & p_{i+k}^0
\end{array}
$$

Définition de ε_i^k

Rappel :

```
dp = p[i+1][k] - 2.*p[i][k] + p[i-1][k];
p[i][k+1] = 2.*p[i][k] - p[i][k-1] + a*dp;
```

Soit ε_i^{k+1} l'erreur commise lors de ces deux lignes de calculs.

On considère a, p_{i-1}^k, p_i^k, p_{i+1}^k et p_i^{k-1} exacts et on regarde l'erreur flottante finale de ces 2 lignes. C'est ε_i^{k+1}.

Définition de ε_i^k

Rappel :

```
dp = p[i+1][k] - 2.*p[i][k] + p[i-1][k];
p[i][k+1] = 2.*p[i][k] - p[i][k-1] + a*dp;
```

Soit ε_i^{k+1} l'erreur commise lors de ces deux lignes de calculs.

On considère a, p_{i-1}^k, p_i^k, p_{i+1}^k et p_i^{k-1} exacts et on regarde l'erreur flottante finale de ces 2 lignes. C'est ε_i^{k+1}.

On sait que les valeurs modèles de $|p_n^m|$ sont majorées par 1.
On suppose que les valeurs flottantes des $|p_n^m|$ sont majorées par 2.

Définition de ε_i^k

Rappel :

```
dp = p[i+1][k] - 2.*p[i][k] + p[i-1][k];
p[i][k+1] = 2.*p[i][k] - p[i][k-1] + a*dp;
```

Soit ε_i^{k+1} l'erreur commise lors de ces deux lignes de calculs.

On considère a, p_{i-1}^k, p_i^k, p_{i+1}^k et p_i^{k-1} exacts et on regarde l'erreur flottante finale de ces 2 lignes. C'est ε_i^{k+1}.

On sait que les valeurs modèles de $|p_n^m|$ sont majorées par 1.
On suppose que les valeurs flottantes des $|p_n^m|$ sont majorées par 2.

$$|\varepsilon_n^m| \leq 85 \times 2^{-52}$$

Définition de ε_i^k

Rappel :

```
dp = p[i+1][k] - 2.*p[i][k] + p[i-1][k];
p[i][k+1] = 2.*p[i][k] - p[i][k-1] + a*dp;
```

Soit ε_i^{k+1} l'erreur commise lors de ces deux lignes de calculs.

On considère a, p_{i-1}^k, p_i^k, p_{i+1}^k et p_i^{k-1} exacts et on regarde l'erreur flottante finale de ces 2 lignes. C'est ε_i^{k+1}.

On sait que les valeurs modèles de $|p_n^m|$ sont majorées par 1.
On suppose que les valeurs flottantes des $|p_n^m|$ sont majorées par 2.

$$|\varepsilon_n^m| \leq 80 \times 2^{-52}$$

Prouvé grâce à la tactique gappa !

Pyramide

L'expression analytique de l'erreur de p_i^k va donc dépendre (et en fait dépendre uniquement) des valeurs de :

$$
\begin{array}{ccccccccccc}
 & & & & \varepsilon_i^k & & & & & & \\
 & & & \varepsilon_{i-1}^{k-1} & & \varepsilon_i^{k-1} & & \varepsilon_{i+1}^{k-1} & & & \\
 & & \varepsilon_{i-2}^{k-2} & & \varepsilon_{i-1}^{k-2} & & \varepsilon_i^{k-2} & & \varepsilon_{i+1}^{k-2} & & \varepsilon_{i+2}^{k-2} \\
 & \iddots & & & & \vdots & & & & \ddots & \\
\varepsilon_{i-k}^0 & & \cdots & & & \varepsilon_i^0 & & & \cdots & & \varepsilon_{i+k}^0
\end{array}
$$

Définition de α_i^k

Étant donné $a \in \mathbb{R}$, je définis $\alpha : \mathbb{Z} \to \mathbb{Z} \to \mathbb{R}$ telle que

$$\alpha_0^0 = 1 \qquad \forall i \neq 0,\ \alpha_i^0 = 0$$

$$\alpha_{-1}^1 = \alpha_1^1 = (1-a) \qquad \alpha_0^1 = 2a \qquad \forall i \notin \{-1, 0, 1\},\ \alpha_i^1 = 0$$

$$\alpha_i^k = a \times (\alpha_{i-1}^{k-1} + \alpha_{i+1}^{k-1}) + 2(1-a) \times \alpha_i^{k-1} - \alpha_i^{k-2}$$

Valeurs des α_i^k

Pour $a = 0.9$, on a :

$$
\begin{array}{ccccccccccccc}
 & & & & & & 1 & & & & & & \\
 & & & & & 0.9 & 0.2 & 0.9 & & & & & \\
 & & & & 0.81 & 0.36 & 0.66 & 0.36 & 0.81 & & & & \\
 & & & 0.729 & 0.486 & 0.495 & 0.58 & 0.495 & 0.486 & 0.729 & & & \\
 & & \iddots & & & & & & & & \ddots & & \\
0.9^k & & & & & & \cdots & & & & & & 0.9^k
\end{array}
$$

Valeurs des α_i^k

Pour $a = 0.9$, on a :

								$\Sigma =$
				1				1
			0.9	0.2	0.9			2
		0.81	0.36	0.66	0.36	0.81		3
	0.729	0.486	0.495	0.58	0.495	0.486	0.729	4
$\ddots$							$\ddots$	
0.9^k				$\cdots$			0.9^k	$k+1$

Valeurs des α_i^k

Pour $a = 0.9$, on a :

$$
\begin{array}{ccccccccccccc}
 & & & & & & 1 & & & & & & \\
 & & & & 0.9 & & 0.2 & & 0.9 & & & & \\
 & & & 0.81 & & 0.36 & & 0.66 & & 0.36 & & 0.81 & \\
 & & 0.729 & & 0.486 & & 0.495 & & 0.58 & & 0.495 & & 0.486 & & 0.729 \\
 & \iddots & & & & & & & & & & & & \ddots & \\
0.9^k & & & & & & & \cdots & & & & & & & 0.9^k
\end{array}
$$

Pour des raisons techniques, j'ai besoin de $\alpha_i^k \geq 0$.

Valeurs des α_i^k

Pour $a = 0.9$, on a :

$$
\begin{array}{ccccccccccccc}
 & & & & & & 1 & & & & & & \\
 & & & & & 0.9 & 0.2 & 0.9 & & & & & \\
 & & & & 0.81 & 0.36 & 0.66 & 0.36 & 0.81 & & & & \\
 & & & 0.729 & 0.486 & 0.495 & 0.58 & 0.495 & 0.486 & 0.729 & & & \\
 & & \iddots & & & & & & & & \ddots & & \\
0.9^k & & & & & & \cdots & & & & & & 0.9^k
\end{array}
$$

Pour des raisons techniques, j'ai besoin de $\alpha_i^k \geq 0$.

F. Clément $\hookrightarrow$ Bruno Salvy $\hookrightarrow$ Manuel Kauers $\hookrightarrow$ Veronika Pillwein

Valeurs des α_i^k

Pour $a = 0.9$, on a :

$$
\begin{array}{ccccccccccc}
 & & & & & 1 & & & & & \\
 & & & & 0.9 & 0.2 & 0.9 & & & & \\
 & & & 0.81 & 0.36 & 0.66 & 0.36 & 0.81 & & & \\
 & & 0.729 & 0.486 & 0.495 & 0.58 & 0.495 & 0.486 & 0.729 & & \\
 & \iddots & & & & & & & & \ddots & \\
0.9^k & & & & & \cdots & & & & & 0.9^k
\end{array}
$$

Pour des raisons techniques, j'ai besoin de $\alpha_i^k \geq 0$.

F. Clément $\hookrightarrow$ Bruno Salvy $\hookrightarrow$ Manuel Kauers $\hookrightarrow$ Veronika Pillwein

$$\alpha_n^j = \sum_{k=j}^{n} \binom{2k}{j+k} \binom{n+k+1}{2k+1} (-1)^{j+k} a^k = a^j \sum_{k=0}^{n-j} P_k^{(2j,0)}(1-2a)$$

Le résultat suit grâce au théorème de Féjer, étendu par Askey et Gaspe.

Expression analytique

En fait, l'erreur de p_i^k est la somme de tous ces gens :

$$\begin{array}{ccccccccc}
 & & & & \varepsilon_i^k & & & & \\
 & & & 0.9\varepsilon_{i-1}^{k-1} & 0.2\varepsilon_i^{k-1} & 0.9\varepsilon_{i+1}^{k-1} & & & \\
 & & 0.81\varepsilon_{i-2}^{k-2} & 0.36\varepsilon_{i-1}^{k-2} & 0.66\varepsilon_i^{k-2} & 0.36\varepsilon_{i+1}^{k-2} & 0.81\varepsilon_{i+2}^{k-2} & & \\
 & \iddots & & & \vdots & & & \ddots & \\
0.9^k\varepsilon_{i-k}^0 & & & & \cdots & & & & 0.9^k\varepsilon_{i+}^0
\end{array}$$

$$p_i^k - exact(p_i^k) = \sum_{l=0}^{k} \sum_{j=-l}^{l} \alpha_j^l \ \varepsilon_{i+j}^{k-l}$$

Expression analytique : conséquences

1. On a une expression analytique de l'erreur d'arrondi.

Expression analytique : conséquences

1. On a une expression analytique de l'erreur d'arrondi.
2. C'est pas si compliqué !
   (on pouvait pas éviter la double sommation pyramidale)

Expression analytique : conséquences

1. On a une expression analytique de l'erreur d'arrondi.
2. C'est pas si compliqué !
   (on pouvait pas éviter la double sommation pyramidale)
3. Ça fait une borne d'erreur en $\bigcirc(k^2\ 2^{-53})$:

$$\left|\mathbf{p_i^k} - \mathbf{exact}\left(\mathbf{p_i^k}\right)\right| \leq \mathbf{85} \times \mathbf{2^{-53}} \times (\mathbf{k}+\mathbf{1}) \times (\mathbf{k}+\mathbf{2})$$

Expression analytique : conséquences

1. On a une expression analytique de l'erreur d'arrondi.
2. C'est pas si compliqué !
   (on pouvait pas éviter la double sommation pyramidale)
3. Ça fait une borne d'erreur en $\bigcirc(k^2\ 2^{-53})$:

$$\left|\mathbf{p_i^k} - \mathbf{exact}\left(\mathbf{p_i^k}\right)\right| \leq \mathbf{85} \times \mathbf{2^{-53}} \times (\mathbf{k}+\mathbf{1}) \times (\mathbf{k}+\mathbf{2})$$

4. C'est pas drôle à prouver formellement.
   Par exemple, $\sum_{l=1}^{k} \sum_{j=-l+1}^{l+1} \alpha_{j-1}^{l}\ \varepsilon_{i+j}^{k-l}$ devient

```
 (sum_f_z (fun l : Z => sum_f_z (fun j : Z =>
alpha a (j - 1) (Zabs_nat l) * eps (i + j) (k -
l)) (- l + 1) (l + 1)) 1 k)%R.
```

   $\Rightarrow$ big operators de ssreflect ?

ε_i^k : les bords

On a la propriété de récurrence :
Si l'erreur est de la forme $\sum\sum\ldots$ aux étapes (i-1,k-1), (i,k-1), (i+1,k-1) et (i,k-2), alors l'erreur est de la forme $\sum\sum\ldots$ à l'étape (i,k).

ε_i^k : les bords

On a la propriété de récurrence :
Si l'erreur est de la forme $\sum\sum\ldots$ aux étapes (i-1,k-1), (i,k-1), (i+1,k-1) et (i,k-2), alors l'erreur est de la forme $\sum\sum\ldots$ à l'étape (i,k).

Problème : les bords ! : $i = 0$ et $i = n_i$.
Là, l'erreur vaut 0, donc n'est pas la somme compliquée qu'on souhaite...

ε_i^k : les bords

On a la propriété de récurrence :
Si l'erreur est de la forme $\sum\sum\ldots$ aux étapes (i-1,k-1), (i,k-1), (i+1,k-1) et (i,k-2), alors l'erreur est de la forme $\sum\sum\ldots$ à l'étape (i,k).

Problème : les bords ! : $i = 0$ et $i = n_i$.
Là, l'erreur vaut 0, donc n'est pas la somme compliquée qu'on souhaite...

sauf si...

ε_i^k : les bords

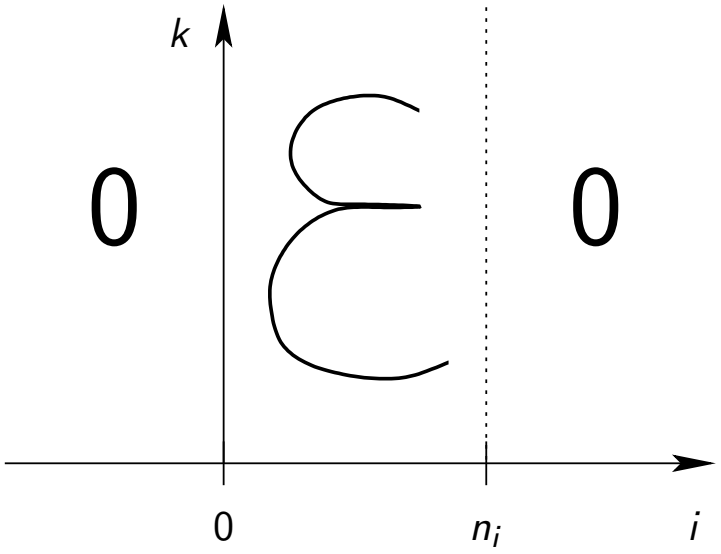
On a la propriété de récurrence :
Si l'erreur est de la forme $\sum\sum\ldots$ aux étapes (i-1,k-1), (i,k-1), (i+1,k-1) et (i,k-2), alors l'erreur est de la forme $\sum\sum\ldots$ à l'étape (i,k).

Problème : les bords ! : $i = 0$ et $i = n_i$.
Là, l'erreur vaut 0, donc n'est pas la somme compliquée qu'on souhaite...

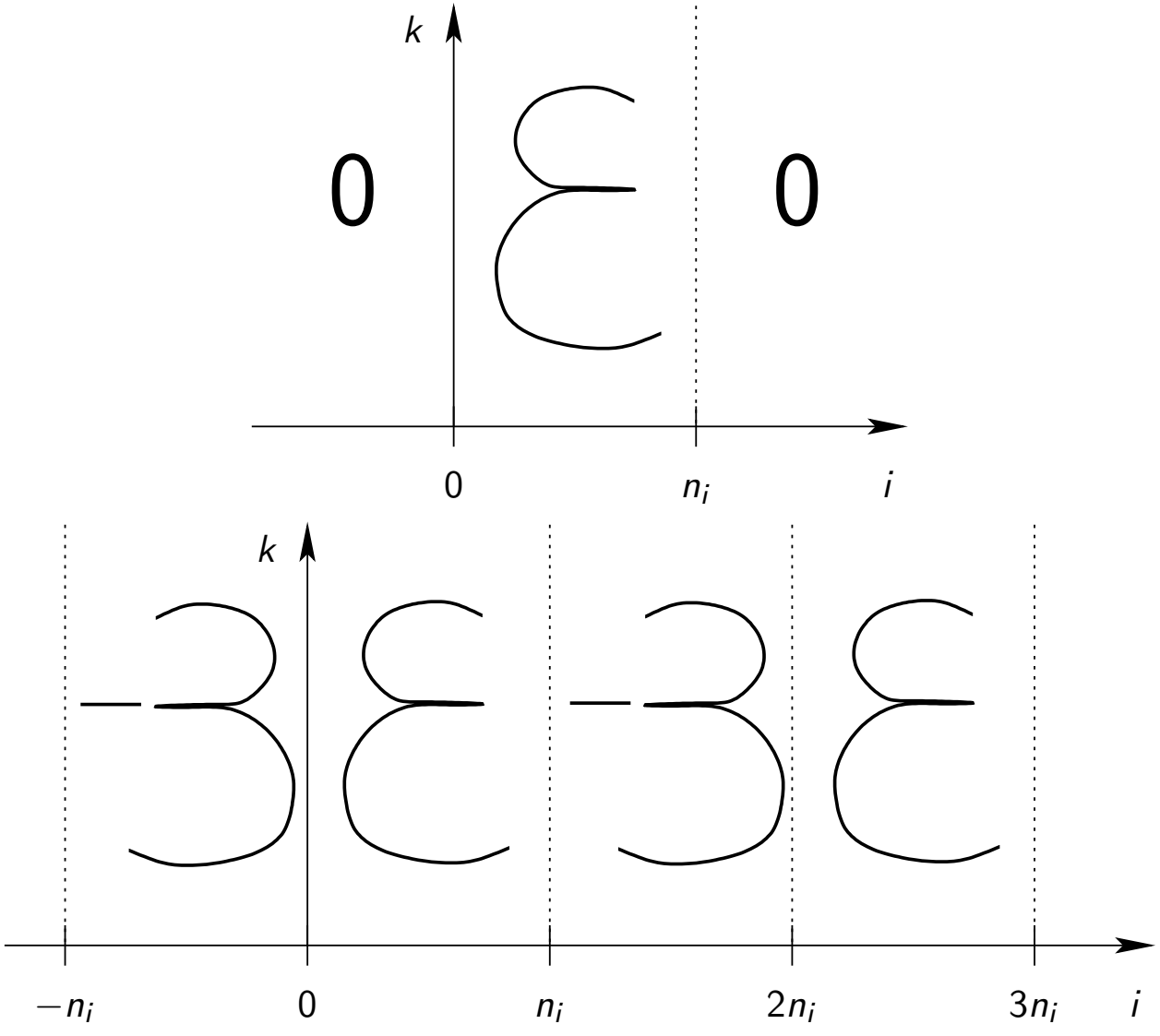
sauf si...

On étend artificiellement les ε_i^k pour $i < 0$ et $i > n_i$ avec les bonnes valeurs (négatives inversées).

Je prends le ε



Je prends le ε et je le retourne



Erreur de méthode

Erreur de méthode (travail en cours !)

Équation des ondes :

$$\frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} = 0$$

$$u \underset{\longrightarrow}{?} u_h \text{ (solution approximée)}$$

$$\text{erreur de méthode} \equiv \text{convergence}$$

$$\text{consistance} \wedge \text{stabilité} \longrightarrow \text{convergence}$$

Consistance

Schéma à 4 point centré explicite :

$$\varepsilon_j^n = \frac{u_h^{n+1} - 2u_h^n + u_h^{n-1}}{\Delta t^2} - c^2 \frac{u_{h+1}^n - 2u_h^n + u_{h-1}^n}{\Delta x^2}$$

Ordre de consistance : 2

$$\varepsilon_j^n = O(\Delta t^2 + \Delta x^2)$$

Stabilité

Sur un intervalle de temps borné :

$$||u_h^n||_{L^2} \leq C(||u_0||_{L^2} + t^n||u_1||_{L^2})$$

Stabilité

Sur un intervalle de temps borné :

$$||u_h^n||_{L^2} \leq C(||u_0||_{L^2} + t^n||u_1||_{L^2})$$

Par des techniques soit :

- de transformée de Fourier
- d'énergie.

Erreur de méthode (travail en cours !)

$\Rightarrow$ on doit redéfinir le produit scalaire, la norme
. . . et les dizaines de lemmes les concernant !

Erreur de méthode (travail en cours !)

$\Rightarrow$ on doit redéfinir le produit scalaire, la norme
... et les dizaines de lemmes les concernant !

$\Rightarrow$ on devrait utiliser $\langle u, v \rangle_{L^2_h} = \sum_{-\infty}^{+\infty} u_j v_j$.
On se ramène à $\langle u, v \rangle_n = \sum_0^n u_j v_j$.

Erreur de méthode (travail en cours !)

$\Rightarrow$ on a besoin de définitions mathématiques “propres”
($f = O(g)$, développements limités, $O(dx^2 + dt^2)\ldots$)

Erreur de méthode (travail en cours !)

$\Rightarrow$ on a besoin de définitions mathématiques “propres”
($f = O(g)$, développements limités, $O(dx^2 + dt^2)\ldots$)

Attention aux échanges implicites entre les quantificateurs existentiels et universels :

$$\forall x, \exists C, P(x, C) \quad \neq \quad \exists C, \forall x, P(x, C)$$

Erreur de méthode (travail en cours !)

$\Rightarrow$ on a besoin de définitions mathématiques "propres"
($f = O(g)$, développements limités, $O(dx^2 + dt^2)\ldots$)

Attention aux échanges implicites entre les quantificateurs existentiels et universels :

$$\forall x, \exists C, P(x, C) \quad \neq \quad \exists C, \forall x, P(x, C)$$

$\Rightarrow$ il semble que la fonction solution doivent être C^n avec ses dérivées n-ièmes bornées pour n "assez grand".

$\Rightarrow$ développement limité uniforme ?

Conclusion

Au boulot !