JNCF'08

Preuves formelles et équation des ondes

Sylvie Boldo

23 octobre 2008

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

centre de recherche SACLAY - ÎLE-DE-FRANCE

JNCF'08

Preuves formelles et équation des ondes

Le Coq et la Corde

Sylvie Boldo

23 octobre 2008

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

centre de recherche SACLAY - ÎLE-DE-FRANCE

Motivations

- On commence à pouvoir spécifier et prouver formellement des programmes avec des nombres flottants.

Motivations

- On commence à pouvoir spécifier et prouver formellement des programmes avec des nombres flottants.
- Regardons de vrais programmes d'analyse numérique

Motivations

- On commence à pouvoir spécifier et prouver formellement des programmes avec des nombres flottants.
- Regardons de vrais programmes d'analyse numérique
- Bon d'accord, on en regarde un facile. . .

Motivations

- On commence à pouvoir spécifier et prouver formellement des programmes avec des nombres flottants.
- Regardons de vrais programmes d'analyse numérique
- Bon d'accord, on en regarde un facile. . .

- Ce travail[1] est en collaboration avec François Clément (INRIA Paris - Rocquencourt), Jean-Christophe Filliâtre (CNRS, LRI) et Micaela Mayero (Université Paris 13).

[1] financé par l'ANR blanche CerPAN (2005–2008)

Plan

Plan

Preuve formelle

La preuve est vérifiée dans ses moindres détails, jusqu'à ce que l'ordinateur l'accepte.

Nous utilisons des assistants de preuves (*formal proof checkers*), c'est-à-dire des programmes qui ne font que vérifier une preuve (et parfois générer une preuve triviale).

En conséquence, le vérificateur est un programme très court (critère de de Bruijn : la correction d'un système entier ne dépend que de la correction d'un très petit « noyau »).

L'assistant de preuves Coq (http://coq.inria.fr)

- basé sur l'isomorphisme de Curry-Howard
  (équivalence entre preuve et λ-terme)
  $\Rightarrow$ garantie théorique
- peu d'automatisations
- des bibliothèques fournies, notamment sur $\mathbb{Z}$ et $\mathbb{R}$
- nombres flottants grâce à L. Théry, M. Daumas et L. Rideau
- Coq vérifie mécaniquement chaque étape de chaque preuve.
- Le but est d'appliquer successivement des tactiques (application de théorème, réécriture, calcul...) pour modifier ou résoudre un but.
- La preuve est faite en partant de la conclusion.

La corde mathématique

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

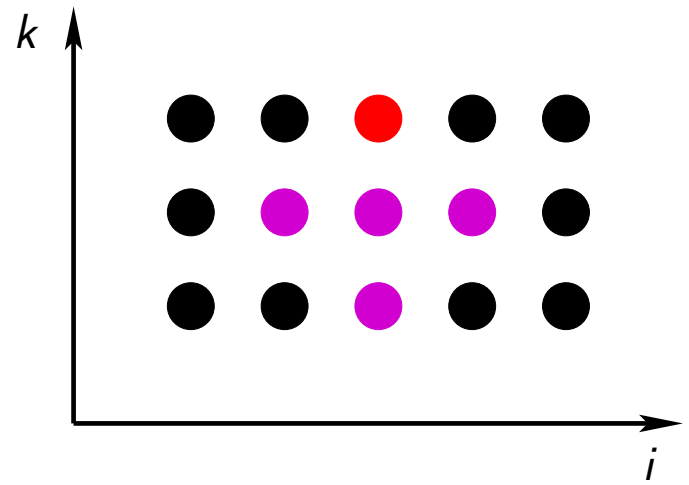
```
[...]  // initialisations de p[i][0] et p[i][1]

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

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.

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.

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

Pyramide

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

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

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

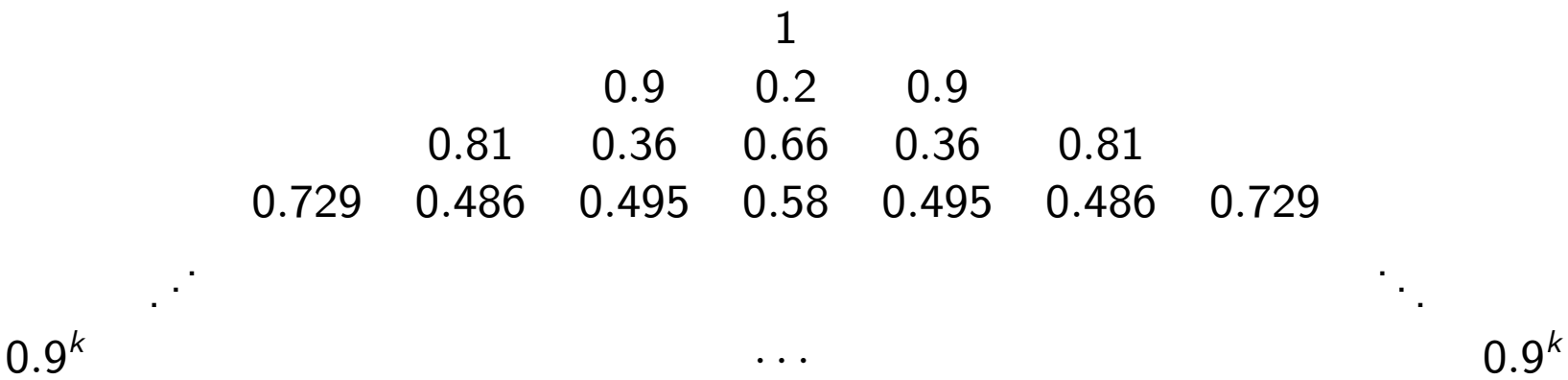
Pour $a = 0.9$, on a :

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

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

Valeurs des α_i^k

Pour $a = 0.9$, on a :

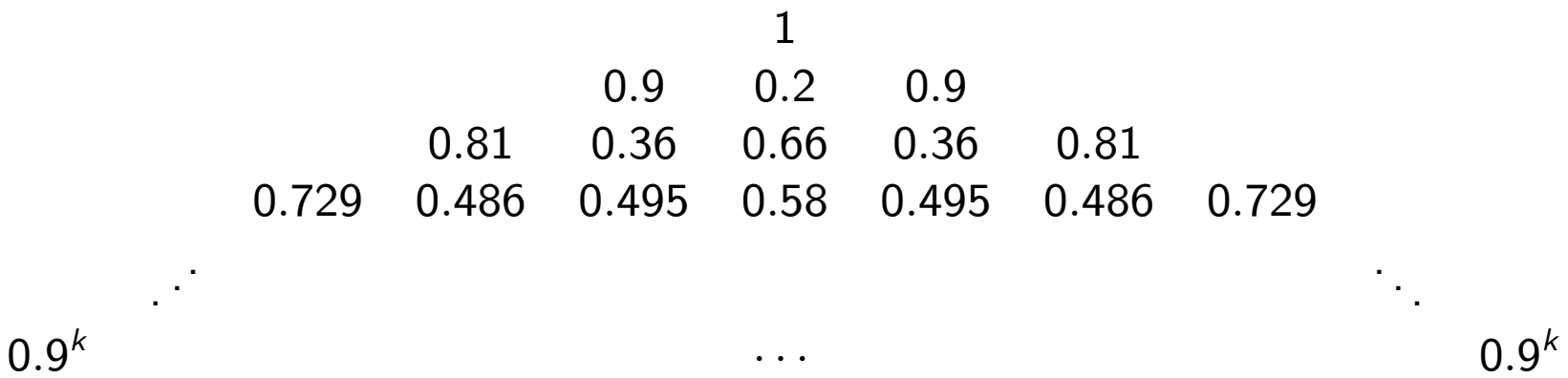


Pour des raisons techniques, j'ai besoin de $\alpha_i^k \geq 0$.

François Clément $\hookrightarrow$ Bruno Salvy $\hookrightarrow$ Manuel Kauers $\hookrightarrow$ Veronika Pillwein

Valeurs des α_i^k

Pour $a = 0.9$, on a :



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

François 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 :

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

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

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.
```

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

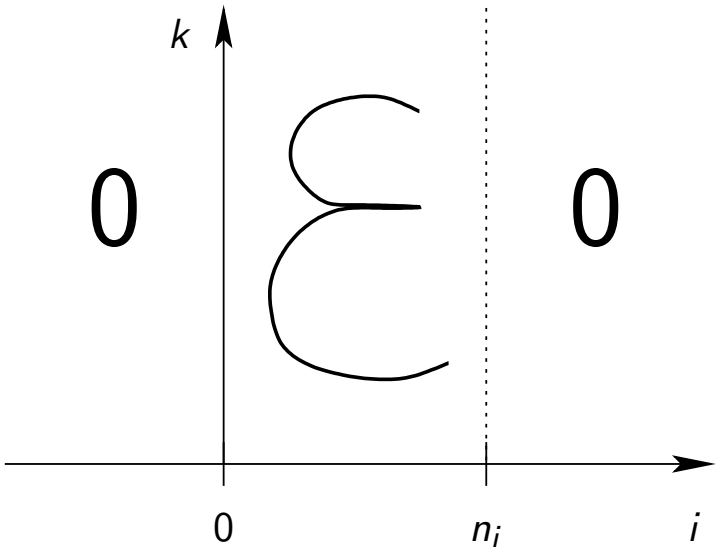
On a la propriété de récurrence :
Si l'erreur est de la forme $\sum\sum\ldots$ aux étapes (i-1,k-1), (i,k-1), (i+1,k-1) et (i,k-2), alors l'erreur est de la forme $\sum\sum\ldots$ à l'étape (i,k).

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

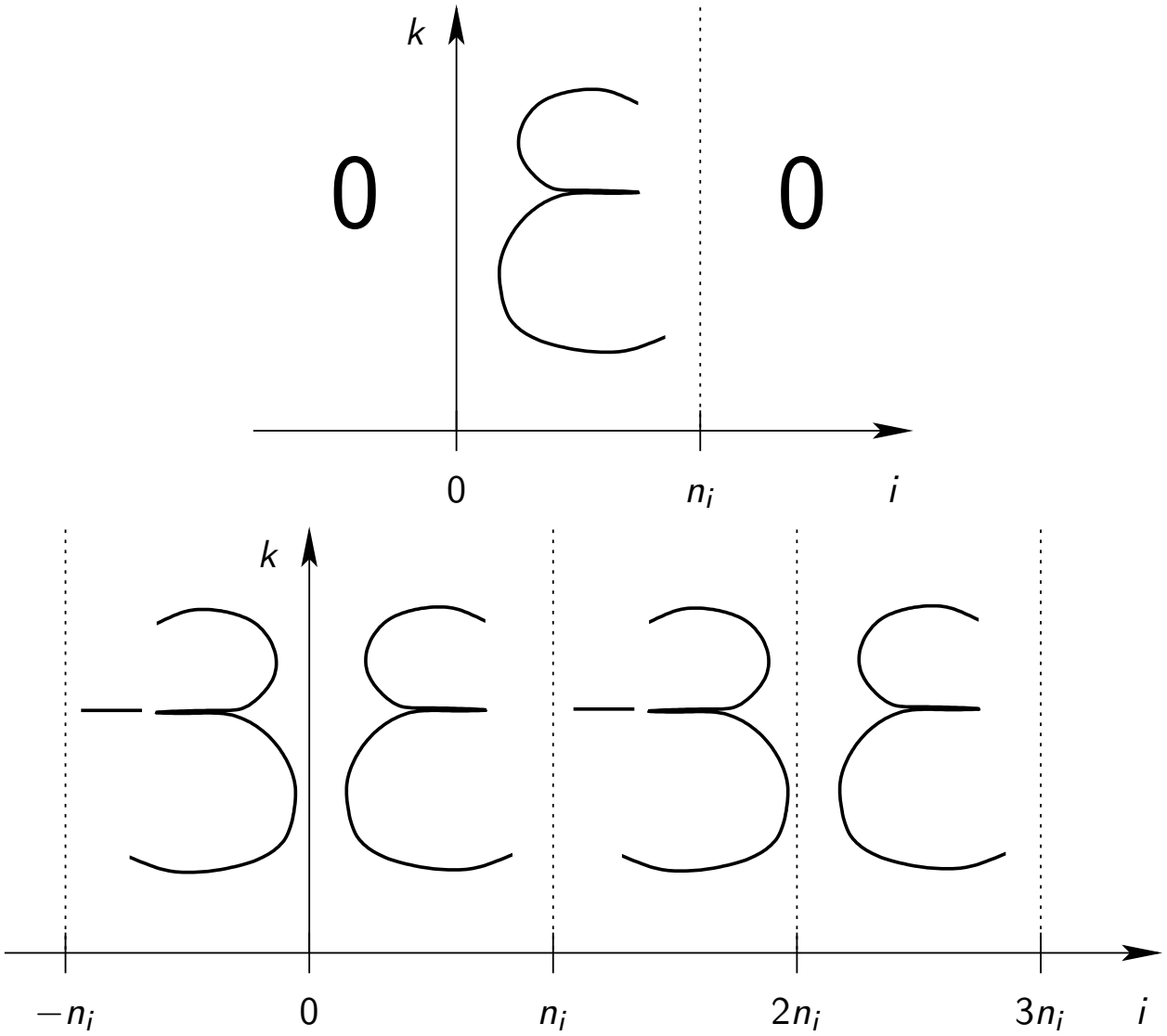
sauf si...

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

On travaille sur l'erreur de méthode mais. . .

Erreur de méthode

On travaille sur l'erreur de méthode mais...

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

Erreur de méthode

On travaille sur l'erreur de méthode mais...

- on a besoin de définitions mathématiques "propres" ($f = O(g)$, développements limités, $O(dx^2 + dt^2)$...),
- si on regarde le point (i, k) qui est à la position $(i \times dx, k \times dt)$ et qu'on fait tendre dt et dx vers 0, alors le point se rapproche de l'origine,

Erreur de méthode

On travaille sur l'erreur de méthode mais...

- on a besoin de définitions mathématiques "propres" ($f = O(g)$, développements limités, $O(dx^2 + dt^2)\ldots$),
- si on regarde le point (i, k) qui est à la position $(i \times dx, k \times dt)$ et qu'on fait tendre dt et dx vers 0, alors le point se rapproche de l'origine,
- 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

On travaille sur l'erreur de méthode mais...

- on a besoin de définitions mathématiques "propres" ($f = O(g)$, développements limités, $O(dx^2 + dt^2)\ldots$),
- si on regarde le point (i, k) qui est à la position $(i \times dx, k \times dt)$ et qu'on fait tendre dt et dx vers 0, alors le point se rapproche de l'origine,
- attention aux échanges implicites entre les quantificateurs existentiels et universels :

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

Bref, c'est plus compliqué et plus long que prévu !

Conclusion (après $\approx$ 2500 loCoq pour 6 loC)

Il reste à

- finir la preuve flottante (quelques initialisations),
- faire la preuve de l'erreur de méthode.

Conclusion (après $\approx$ 2500 loCoq pour 6 loC)

Il reste à

- finir la preuve flottante (quelques initialisations),
- faire la preuve de l'erreur de méthode.

Ça nous a appris que

- Les preuves formelles, c'est bien, car ça augmente la confiance et ça trouve les erreurs ou les flous dans les démonstrations.
- Entre les preuves des numériciens et les preuves formelles, il y a un gouffre très profond.