Accélération de calcul du point-fixe en interprétation abstraite en utilisant la convergence des suites numériques.

Yassamine Seladji

CEA, LIST, Laboratoire de Modélisation et d'Analyse des Systemes en Intéraction.
91191 Gif-sur-Yvette, France

Séminaire LSL.
15 décembre 2010

Plan

Accélération de calcul du point-fixe en interprétation abstraite en utilisant la convergence des suites numériques.

Plan

Accélération de calcul du point-fixe en interprétation abstraite en utilisant la convergence des suites numériques.

2. Calcul du point-fixe
   - L'itération de Kleene
   - Le widening
   - Le widening avec seuils

Plan

Accélération de calcul du point-fixe en interprétation abstraite en utilisant la convergence des suites numériques.

3. Méthodes d'accélération de convergence en analyse numérique
   - Accélération de convergence des suites numériques
   - La méthode Δ^2-d'Aitken
   - ε-Algorithme
   - ε-Algorithme Vectoriel

Plan

Accélération de calcul du point-fixe en interprétation abstraite en utilisant la convergence des suites numériques.

Plan

Accélération de calcul du point-fixe en interprétation abstraite en utilisant la convergence des suites numériques.

Le contexte.

Les problématiques industrielles

- Le crash d'Ariane 5: dépassement de capacité.
  $\Longrightarrow$ 700 Millions d'euros de perte.
- La panne des moteurs de USS Yorktown CG: division par zéro.
  $\Longrightarrow$ arrêt forcé et 1 Million de Dollars de perte.
- L'erreur d'indice boursier à Vancouver: erreur d'arrondi.
  $\Longrightarrow$ Indice erroné, perte d'argent.

Une approche intuitive.

La méthode naive:

L'idée.

Pour verifier la sûreté d'un programme, il suffit de déterminer de manière exhaustive les comportements du programme pour tout l'ensemble des entrées. Et ainsi détécter les comportements indésirables.

Une approche intuitive.

La méthode naive:

L'idée.

Pour verifier la sûreté d'un programme, il suffit de déterminer de manière exhaustive les comportements du programme pour tout l'ensemble des entrées. Et ainsi détécter les comportements indésirables.

Exemple

1: x $\in\{90, \ldots, 100\}$;
2: While $(x \geq 0)$
3:{
4: y := $f(x)/x$;$\Rightarrow$ Error
5: x := x-1;
6: }
7:

$\Longrightarrow$

Les valeurs de X

Itération:101

	x
1	$\{90, \ldots, 100\}$
2	$\{0, \ldots, 100\}$
5	$\{-1, \ldots, 99\}$
7	$\{-1\}$

Une approche intuitive.

Les limites de la méthode naive :

- Difficilement applicable aux gros programmes.
  - Explosion de la taille de l'ensemble des comportements: problème de la représentation des données en mémoire.
  - Pour les programmes avec boucles: possibilité de non-terminaison du calcul.

Une approche intuitive.

Les limites de la méthode naive :

- Difficilement applicable aux gros programmes.
  - Explosion de la taille de l'ensemble des comportements: problème de la représentation des données en mémoire.
  - Pour les programmes avec boucles: possibilité de non-terminaison du calcul.

Une solution: analyse statique par interprétation abstraite.

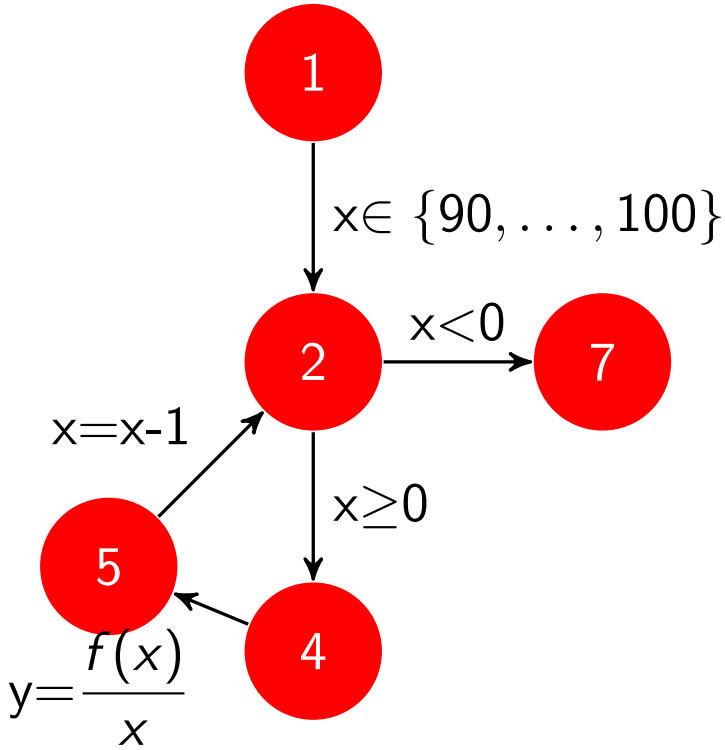
L'approche formelle

Le code source

1: $x \in \{90, \ldots, 100\}$;
2: While $(x \geq 0)$
3:{
4: $y := f(x)/x$;
5: x := x-1;
6: }
7:

$\Longrightarrow$

Graphe de flôt de contrôle



Les équations sémantiques

- $S_1 = \bot$.

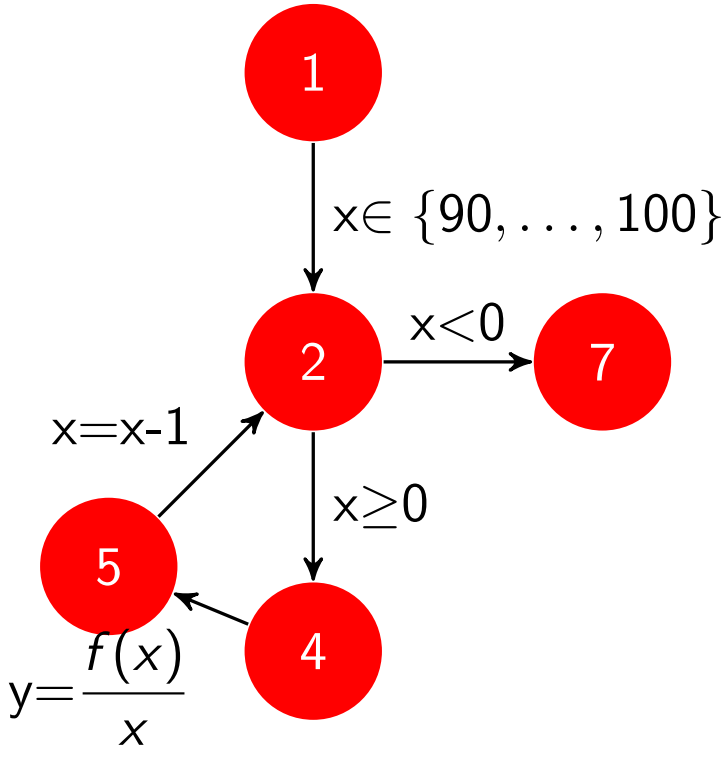
L'approche formelle

Le code source

1: $x \in \{90, \ldots, 100\}$;
2: While $(x \geq 0)$
3:{
4: y := $f(x)/x$;
5: x := x-1;
6: }
7:

$\Longrightarrow$

Graphe de flôt de contrôle



Les équations sémantiques

- $S_1 = \bot$.
- $S_2 = [\![x \in \{90, \ldots, 100\}]\!](S_1) \cup [\![x = x - 1]\!](S_5)$.

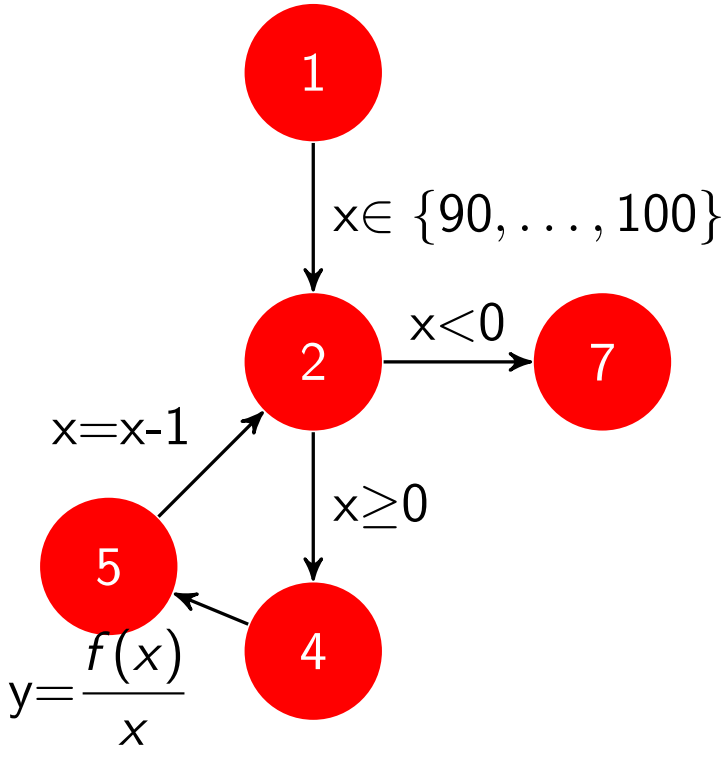
L'approche formelle

Le code source

1: x $\in \{90, \ldots, 100\}$;
2: While $(x \geq 0)$
3:{
4: y := $f(x)/x$;
5: x := x-1;
6: }
7:

$\Longrightarrow$

Graphe de flôt de contrôle



Les équations sémantiques

- $S_1 = \bot$.
- $S_2 = [\![x \in \{90, \ldots, 100\}]\!](S_1) \cup [\![x = x - 1]\!](S_5)$.
- $S_4 = [\![x \geq 0]\!](S_2)$.

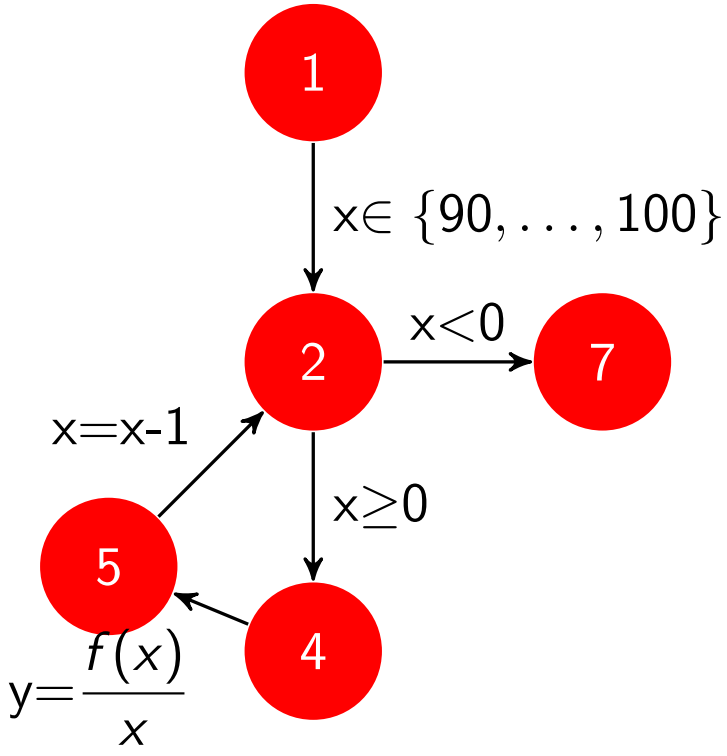
L'approche formelle

Le code source

```
1: x ∈ {90, . . . , 100};
2: While (x ≥ 0)
3:{
4: y := f(x)/x;
5: x := x-1;
6: }
7:
```

⟹

Graphe de flôt de contrôle



Les équations sémantiques

- $S_1 = \bot$.
- $S_2 = [\![x \in \{90, \ldots, 100\}]\!](S_1) \cup [\![x = x - 1]\!](S_5)$.
- $S_4 = [\![x \geq 0]\!](S_2)$.
- $S_5 = [\![y = \frac{f(x)}{x}]\!](S_4)$.

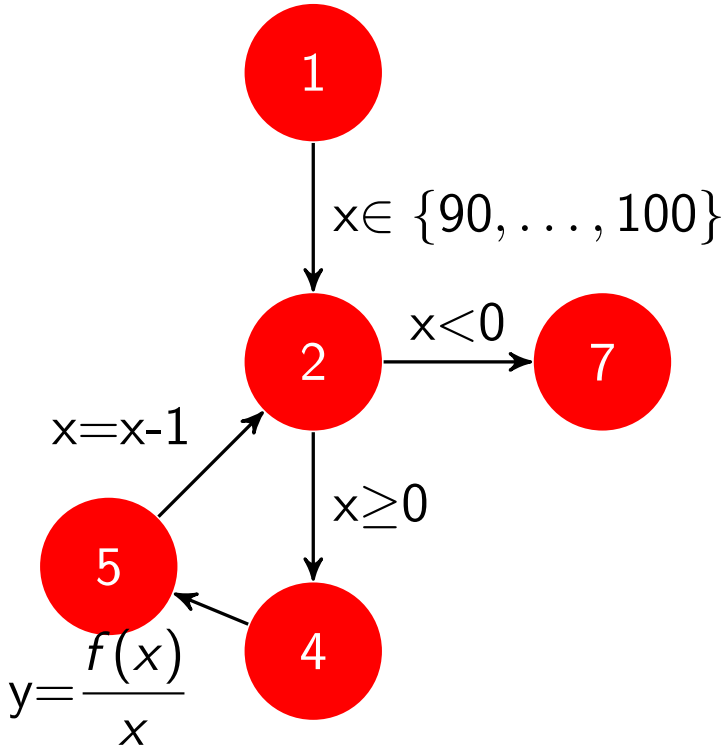
L'approche formelle

Le code source

```
1: x ∈ {90, . . . , 100};
2: While (x ≥ 0)
3:{
4: y := f(x)/x;
5: x := x-1;
6: }
7:
```

⟹

Graphe de flôt de contrôle



Les équations sémantiques

- $S_1 = \bot$.
- $S_2 = [\![x \in \{90, \ldots, 100\}]\!](S_1) \cup [\![x = x - 1]\!](S_5)$.
- $S_4 = [\![x \geq 0]\!](S_2)$.
- $S_5 = [\![y = \frac{f(x)}{x}]\!](S_4)$. $S_7 = [\![x < 0]\!](S_2)$.

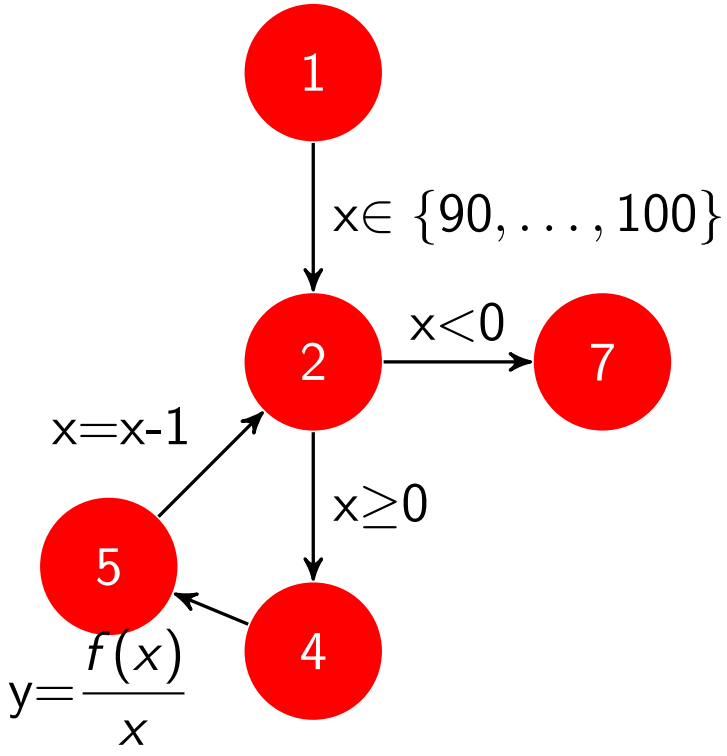
L'approche formelle

Le code source

1: $x \in \{90, \ldots, 100\}$;
2: While $(x \geq 0)$
3:{
4: $y := f(x)/x$;
5: x := x-1;
6: }
7:

$\Longrightarrow$

Graphe de flôt de contrôle



Les équations sémantiques

- $S_1 = \bot$.
- $S_2 = [\![x \in \{90, \ldots, 100\}]\!](S_1) \cup [\![x = x - 1]\!](S_5)$.
- $S_4 = [\![x \geq 0]\!](S_2)$.
- $S_5 = [\![y = \frac{f(x)}{x}]\!](S_4)$. $S_7 = [\![x < 0]\!](S_2)$.

$$\vec{S} = F(\vec{S}).$$

L'approche formelle[Cousot/Cousot77]

- L'Analyse statique par interprétation abstraite consiste à déterminer une sur-approximation des ensembles de comportements du programme sans l'exécuter.
- La détermination de l'ensemble des comportements passe par le calcul du plus petit point-fixe de F.
- **Les solutions apportées:**
  - Non-terminaison du calcul
  - La représentation des données en mémoire

L'approche formelle[Cousot/Cousot77]

- L'Analyse statique par interprétation abstraite consiste à déterminer une sur-approximation des ensembles de comportements du programme sans l'exécuter.
- La détermination de l'ensemble des comportements passe par le calcul du plus petit point-fixe de F.
- **Les solutions apportées:**
  - Non-terminaison du calcul $\Longrightarrow$ Widening.
  - La représentation des données en mémoire

L'approche formelle[Cousot/Cousot77]

- L'Analyse statique par interprétation abstraite consiste à déterminer une sur-approximation des ensembles de comportements du programme sans l'exécuter.
- La détermination de l'ensemble des comportements passe par le calcul du plus petit point-fixe de F.
- **Les solutions apportées:**
  - Non-terminaison du calcul $\Longrightarrow$ Widening.
  - La représentation des données en mémoire $\Longrightarrow$ Domaine abstrait.

Les domaines abstraits

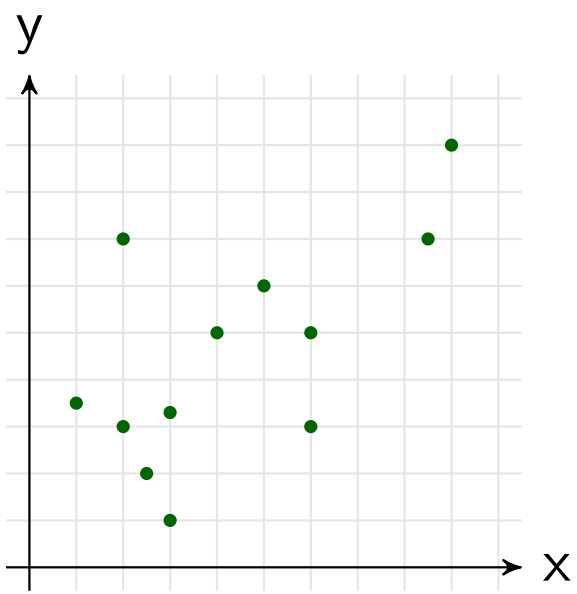
Les domaines abstraits non-relationnels

Un domaine abstrait non-relationnel est un ensemble ordonné d'éléments permettant de représenter chaque variable indépendament des autres variables du programme. Exemple: Domaine des intervalles.
Pour $D = \{1, 4, 8, 9, 20\} \Longrightarrow D_I = [1, 20]$

Les domaines abstraits

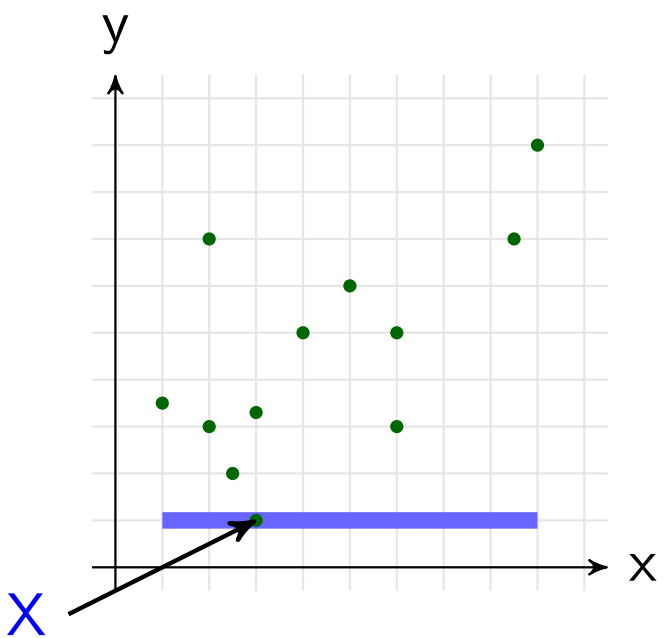
Les domaines abstraits non-relationnels

Un domaine abstrait non-relationnel est un ensemble ordonné d'éléments permettant de représenter chaque variable indépendamment des autres variables du programme. Exemple: Domaine des intervalles.
Pour $D = \{1, 4, 8, 9, 20\} \Longrightarrow D_I = [1, 20]$



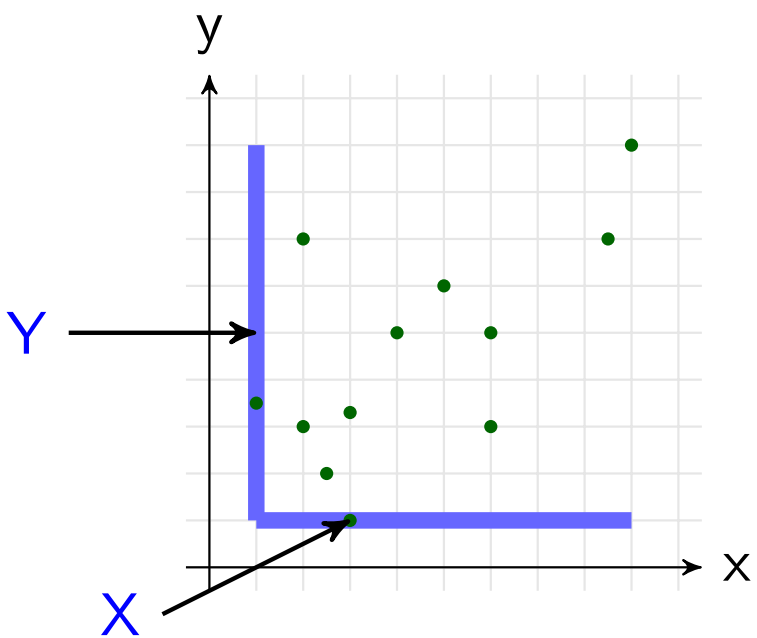
Les domaines abstraits non-relationnels

Un domaine abstrait non-relationnel est un ensemble ordonné d'éléments permettant de représenter chaque variable indépendamment des autres variables du programme. Exemple: Domaine des intervalles.
Pour $D = \{1, 4, 8, 9, 20\} \Longrightarrow D_I = [1, 20]$



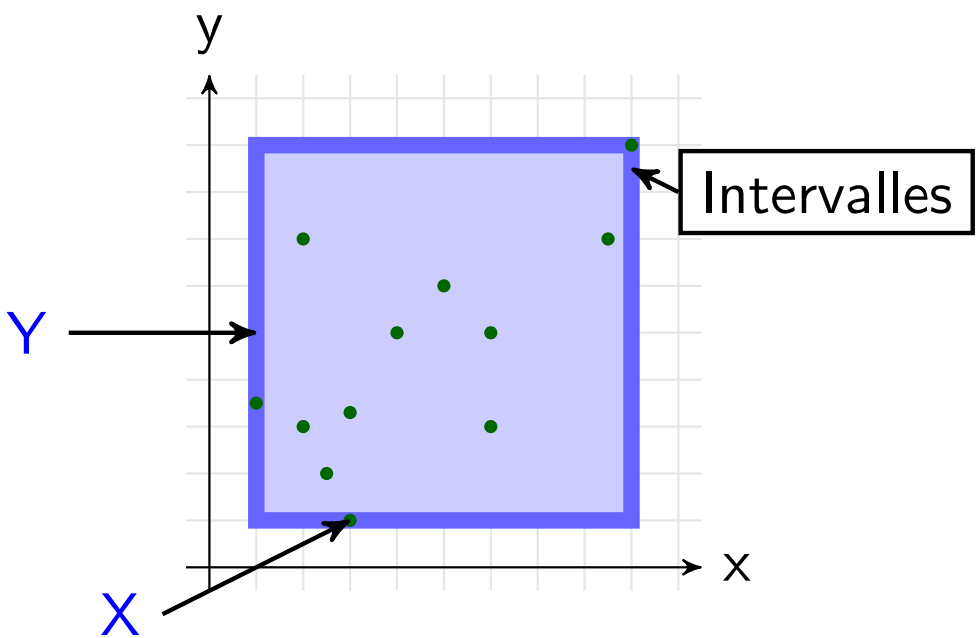
Les domaines abstraits non-relationnels

Un domaine abstrait non-relationnel est un ensemble ordonné d'éléments permettant de représenter chaque variable indépendamment des autres variables du programme. Exemple: Domaine des intervalles.
Pour $D = \{1, 4, 8, 9, 20\} \Longrightarrow D_I = [1, 20]$



Les domaines abstraits non-relationnels

Un domaine abstrait non-relationnel est un ensemble ordonné d'éléments permettant de représenter chaque variable indépendamment des autres variables du programme. Exemple: Domaine des intervalles.
Pour $D = \{1, 4, 8, 9, 20\} \Longrightarrow D_I = [1, 20]$



Les domaines abstraits relationnels

Un domaine abstrait relationnel est un ensemble ordonné d'éléments qui permet de représenter les variables en prenant en compte les relations existantes entre eux dans le programme.

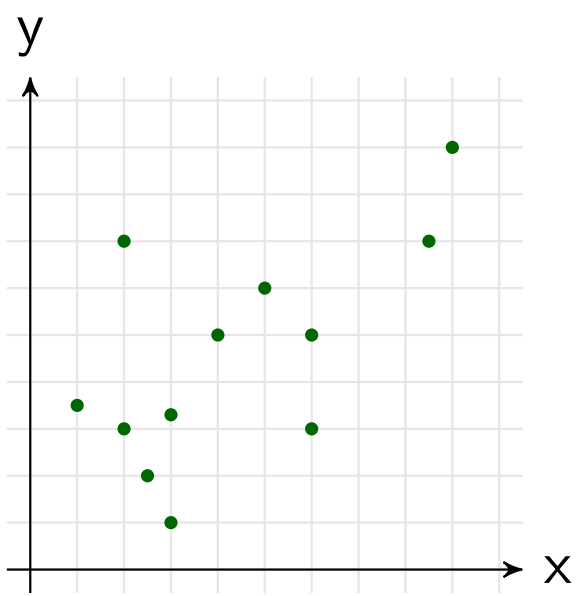
Exemple: le domaine des polyèdres: les relations entre les variables sont représentées par des contraintes du type: $\sum_{x_i \in Var} \alpha_i x_i \leq C$.

Les domaines abstraits

Les domaines abstraits relationnels

Un domaine abstrait relationnel est un ensemble ordonné d'éléments qui permet de représenter les variables en prenant en compte les relations existantes entre eux dans le programme.

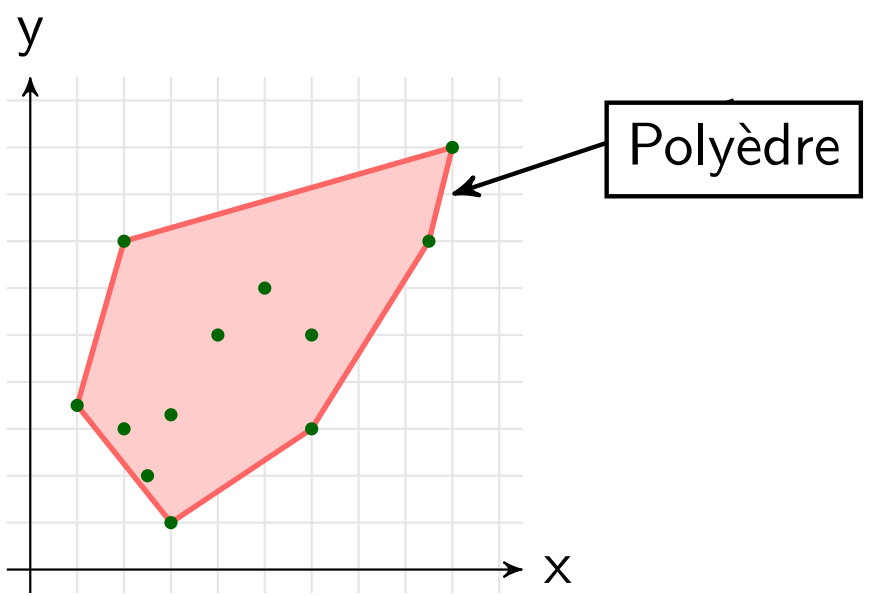
Exemple: le domaine des polyèdres: les relations entre les variables sont représentées par des contraintes du type: $\sum_{x_i \in Var} \alpha_i x_i \leq C$.



Les domaines abstraits relationnels

Un domaine abstrait relationnel est un ensemble ordonné d'éléments qui permet de représenter les variables en prenant en compte les relations existantes entre eux dans le programme.

Exemple: le domaine des polyèdres: les relations entre les variables sont représentées par des contraintes du type: $\sum_{x_i \in Var} \alpha_i x_i \leq C$.



Plan

Accélération de calcul du point-fixe en interprétation abstraite en utilisant la convergence des suites numériques.

L'itération de Kleene

Algorithm 1 Méthode d'itération de Kleene

1: $\overrightarrow{X}_i := \bot$
2: **repeat**
3: $\quad \overrightarrow{X}_i := \overrightarrow{X}_{i-1} \sqcup F(\overrightarrow{X}_{i-1})$
4: **until** $\overrightarrow{X}_i \sqsubseteq \overrightarrow{X}_{i-1}$

Exemple

1: x = [90,100];
2: While $(x \geq 0)$
3:{
4: y := $f(x)$/x; $\Longrightarrow$ Error
5: x := x-1;
6: }
7:

$\Longrightarrow$

Calcul du point-fixe

Itération: 101

	x
1	[90,100]
2	[0,100]
5	[-1,99]
7	[-1,-1]

Le widening

Algorithm 2 Méthode d'itération de Kleene avec Widening

1: $\overrightarrow{X}_i := \bot$
2: **repeat**
3: $\quad \overrightarrow{X}_i := \overrightarrow{X}_{i-1} \nabla F(\overrightarrow{X}_{i-1})$
4: **until** $\overrightarrow{X}_i \sqsubseteq \overrightarrow{X}_{i-1}$

Définition

Soit $(D, \sqsubseteq)$ un ensemble ordonné. ∇ est un opérateur de $D \to D$ tel que:

- *i)* $\forall x, y \in D$, $x \sqcup y \sqsubseteq x \nabla y$.
- *ii)* Pour une suite croissante (X_n) d'éléments de D, la nouvelle suite (Y_n) avec $Y_n = Y_{n-1} \nabla X_n$ est ultimement stationnaire.

Le widening

Algorithm 3 Méthode d'itération de Kleene avec Widening

1: $\overrightarrow{X}_i := \bot$
2: **repeat**
3: $\quad \overrightarrow{X}_i := \overrightarrow{X}_{i-1} \nabla F(\overrightarrow{X}_{i-1})$
4: **until** $\overrightarrow{X}_i \sqsubseteq \overrightarrow{X}_{i-1}$

Exemple: Domaine des intervalles

$$[a, b] \nabla [c, d] = \left[\begin{cases} a & \text{Si } a \leq c \\ -\infty & \text{sinon} \end{cases}, \begin{cases} b & \text{Si } b \geq d \\ +\infty & \text{sinon} \end{cases} \right].$$

Le widening

Algorithm 4 Méthode d'itération de Kleene avec Widening

1: $\overrightarrow{X}_i := \bot$
2: **repeat**
3: $\quad \overrightarrow{X}_i := \overrightarrow{X}_{i-1} \nabla F(\overrightarrow{X}_{i-1})$
4: **until** $\overrightarrow{X}_i \sqsubseteq \overrightarrow{X}_{i-1}$

Exemple

1: x = [90,100];
2: While $(x \geq 0)$
3:{
4: y := $f(x)$/x;
5: x := x-1;
6: }
7:

$\Longrightarrow$

Calcul du point-fixe

Itération:1

	x
1	[90,100]
2	[90,100]
5	[89,99]
7	$\bot$

Le widening

Algorithm 5 Méthode d'itération de Kleene avec Widening

1: $\overrightarrow{X}_i := \bot$
2: **repeat**
3: $\quad \overrightarrow{X}_i := \overrightarrow{X}_{i-1} \nabla F(\overrightarrow{X}_{i-1})$
4: **until** $\overrightarrow{X}_i \sqsubseteq \overrightarrow{X}_{i-1}$

Exemple

1: x = [90,100];
2: While $(x \geq 0)$
3:{
4: y := $f(x)$/x;
5: x := x-1;
6: }
7:

$\Longrightarrow$

Calcul du point-fixe

Itération:1

	x
1	[90,100]
2	[89,100]
5	[89,99]
7	$\bot$

Le widening

Algorithm 6 Méthode d'itération de Kleene avec Widening

1: $\overrightarrow{X}_i := \bot$
2: **repeat**
3: $\quad \overrightarrow{X}_i := \overrightarrow{X}_{i-1} \nabla F(\overrightarrow{X}_{i-1})$
4: **until** $\overrightarrow{X}_i \sqsubseteq \overrightarrow{X}_{i-1}$

Exemple

1: x = [90,100];
2: While $(x \geq 0)$
3:{
4: y := $f(x)$/x;
5: x := x-1;
6: }
7:

$\Longrightarrow$

Calcul du point-fixe

Itération:2

	x
1	[90,100]
2	[89,100]
5	[88,99]
7	$\bot$

Le widening

Algorithm 7 Méthode d'itération de Kleene avec Widening

1: $\overrightarrow{X}_i := \bot$
2: **repeat**
3: $\quad \overrightarrow{X}_i := \overrightarrow{X}_{i-1} \nabla F(\overrightarrow{X}_{i-1})$
4: **until** $\overrightarrow{X}_i \sqsubseteq \overrightarrow{X}_{i-1}$

Exemple

1: x = [90,100];
2: While $(x \geq 0)$
3:{
4: y := $f(x)$/x; $\Longrightarrow$ Error
5: x := x-1;
6: }
7:

$\Longrightarrow$

Calcul du point-fixe

Itération:2

	x
1	[90,100]
2	$[-\infty$,100]
5	[88,99]
7	$\bot$

Le widening

Algorithm 8 Méthode d'itération de Kleene avec Widening

1: $\overrightarrow{X}_i := \bot$
2: **repeat**
3: $\quad \overrightarrow{X}_i := \overrightarrow{X}_{i-1} \nabla F(\overrightarrow{X}_{i-1})$
4: **until** $\overrightarrow{X}_i \sqsubseteq \overrightarrow{X}_{i-1}$

Exemple

1: x = [90,100];
2: While $(x \geq 0)$
3:{
4: y := $f(x)$/x; $\Longrightarrow$ Error
5: x := x-1;
6: }
7:

$\Longrightarrow$

Calcul du point-fixe

Itération:3

	x
1	[90,100]
2	$[-\infty,100]$
5	$[-1,99]$
7	$\bot$

Le widening

Algorithm 9 Méthode d'itération de Kleene avec Widening

1: $\overrightarrow{X}_i := \bot$
2: **repeat**
3: $\quad \overrightarrow{X}_i := \overrightarrow{X}_{i-1} \nabla F(\overrightarrow{X}_{i-1})$
4: **until** $\overrightarrow{X}_i \sqsubseteq \overrightarrow{X}_{i-1}$

Exemple

1: x = [90,100];
2: While $(x \geq 0)$
3:{
4: y := $f(x)$/x; $\Longrightarrow$ Error
5: x := x-1;
6: }
7:

$\Longrightarrow$

Calcul du point-fixe

Itération:3

	x
1	[90,100]
2	$[-\infty$,100]
5	[−1,99]
7	$[-\infty$,-1]

Le widening

Algorithm 10 Méthode d'itération de Kleene avec Widening

1: $\overrightarrow{X}_i := \bot$
2: **repeat**
3: $\quad \overrightarrow{X}_i := \overrightarrow{X}_{i-1} \nabla F(\overrightarrow{X}_{i-1})$
4: **until** $\overrightarrow{X}_i \sqsubseteq \overrightarrow{X}_{i-1}$

- Avantages:
  - Forcer la terminaison du calcul.
  - Permet un calcul rapide.
- Inconvénients:
  - Perdre la précision des calculs.
  - Perdre la notion du plus petit point-fixe.

Le widening avec seuils

Algorithm 11 Méthode d'itération de Kleene avec widening avec seuils

1: $\overrightarrow{X}_i := \bot$
2: **repeat**
3: $\quad \overrightarrow{X}_i := \overrightarrow{X}_{i-1} \nabla_T F(\overrightarrow{X}_{i-1}) \Longrightarrow \{50, 10, 0\}$.
4: **until** $\overrightarrow{X}_i \sqsubseteq \overrightarrow{X}_{i-1}$

$$[a,b]\nabla_T[c,d] = [e,f] \text{ avec } e = \left[\begin{cases} 50 & \text{Si } 50 \leq c \leq a \\ 10 & \text{Si } 10 \leq c \leq a \leq 90 \\ 0 & \text{Si } 0 \leq c \leq a \leq 10 \\ -\infty & \text{Si } c \leq a \leq 0 \\ a & \text{sinon} \end{cases} \right.$$

Le widening avec seuils

Algorithm 12 Méthode d'itération de Kleene avec widening avec seuils

1: $\overrightarrow{X}_i := \bot$
2: **repeat**
3: $\quad \overrightarrow{X}_i := \overrightarrow{X}_{i-1} \nabla_T F(\overrightarrow{X}_{i-1}) \Longrightarrow \{50, 10, 0\}$.
4: **until** $\overrightarrow{X}_i \sqsubseteq \overrightarrow{X}_{i-1}$

Exemple

1: x = [90,100];
2: While $(x \geq 0)$
3:{
4: y := $f(x)$/x;
5: x := x-1;
6: }
7:

$\Longrightarrow$

Calcul d'un point-fixe

Itération:1

	x
1	[90,100]
2	[90,100]
5	[89,99]
7	$\bot$

Le widening avec seuils

Algorithm 13 Méthode d'itération de Kleene avec widening avec seuils

1: $\overrightarrow{X}_i := \bot$
2: **repeat**
3: $\quad \overrightarrow{X}_i := \overrightarrow{X}_{i-1} \nabla_T F(\overrightarrow{X}_{i-1}) \Longrightarrow \{50, 10, 0\}$.
4: **until** $\overrightarrow{X}_i \sqsubseteq \overrightarrow{X}_{i-1}$

Exemple

1: x = [90,100];
2: While $(x \geq 0)$
3:{
4: y := $f(x)$/x;
5: x := x-1;
6: }
7:

$\Longrightarrow$

Calcul d'un point-fixe

Itération:1

	x
1	[90,100]
2	[89,100]
5	[89,99]
7	$\bot$

Le widening avec seuils

Algorithm 14 Méthode d'itération de Kleene avec widening avec seuils

1: $\overrightarrow{X}_i := \bot$
2: **repeat**
3: $\quad \overrightarrow{X}_i := \overrightarrow{X}_{i-1} \nabla_T F(\overrightarrow{X}_{i-1}) \Longrightarrow \{50, 10, 0\}$.
4: **until** $\overrightarrow{X}_i \sqsubseteq \overrightarrow{X}_{i-1}$

Exemple

1: x = [90,100];
2: While $(x \geq 0)$
3:{
4: y := $f(x)$/x;
5: x := x-1;
6: }
7:

$\Longrightarrow$

Calcul d'un point-fixe

Itération:2

	x
1	[90,100]
2	[89,100]
5	[88,99]
7	$\bot$

Le widening avec seuils

Algorithm 15 Méthode d'itération de Kleene avec widening avec seuils

1: $\overrightarrow{X}_i := \bot$
2: **repeat**
3: $\quad \overrightarrow{X}_i := \overrightarrow{X}_{i-1} \nabla_T F(\overrightarrow{X}_{i-1}) \Longrightarrow \{50, 10, 0\}$.
4: **until** $\overrightarrow{X}_i \sqsubseteq \overrightarrow{X}_{i-1}$

Exemple

1: x = [90,100];
2: While $(x \geq 0)$
3:{
4: y := $f(x)$/x;
5: x := x-1;
6: }
7:

$\Longrightarrow$

Calcul d'un point-fixe

Itération:2

	x
1	[90,100]
2	[50,100]
5	[88,99]
7	$\bot$

Le widening avec seuils

Algorithm 16 Méthode d'itération de Kleene avec widening avec seuils

1: $\overrightarrow{X}_i := \bot$
2: **repeat**
3: $\quad \overrightarrow{X}_i := \overrightarrow{X}_{i-1} \nabla_T F(\overrightarrow{X}_{i-1}) \Longrightarrow \{50, 10, 0\}$.
4: **until** $\overrightarrow{X}_i \sqsubseteq \overrightarrow{X}_{i-1}$

Exemple

1: x = [90,100];
2: While $(x \geq 0)$
3:{
4: y := $f(x)$/x;
5: x := x-1;
6: }
7:

$\Longrightarrow$

Calcul d'un point-fixe

Itération:3

	x
1	[90,100]
2	[50,100]
5	[49,99]
7	$\bot$

Le widening avec seuils

Algorithm 17 Méthode d'itération de Kleene avec widening avec seuils

1: $\overrightarrow{X}_i := \bot$
2: **repeat**
3: $\quad \overrightarrow{X}_i := \overrightarrow{X}_{i-1} \nabla_T F(\overrightarrow{X}_{i-1}) \Longrightarrow \{50, 10, 0\}$.
4: **until** $\overrightarrow{X}_i \sqsubseteq \overrightarrow{X}_{i-1}$

Exemple

1: x = [90,100];
2: While $(x \geq 0)$
3:{
4: y := $f(x)$/x;
5: x := x-1;
6: }
7:

$\Longrightarrow$

Calcul d'un point-fixe

Itération:3

	x
1	[90,100]
2	[10,100]
5	[49,99]
7	$\bot$

Le widening avec seuils

Algorithm 18 Méthode d'itération de Kleene avec widening avec seuils

1: $\overrightarrow{X}_i := \bot$
2: **repeat**
3: $\quad \overrightarrow{X}_i := \overrightarrow{X}_{i-1} \nabla_T F(\overrightarrow{X}_{i-1}) \Longrightarrow \{50, 10, 0\}$.
4: **until** $\overrightarrow{X}_i \sqsubseteq \overrightarrow{X}_{i-1}$

Exemple

1: x = [90,100];
2: While $(x \geq 0)$
3:{
4: y := $f(x)$/x;
5: x := x-1;
6: }
7:

$\Longrightarrow$

Calcul d'un point-fixe

Itération:4

	x
1	[90,100]
2	[10,100]
5	[9,99]
7	$\bot$

Le widening avec seuils

Algorithm 19 Méthode d'itération de Kleene avec widening avec seuils

1: $\overrightarrow{X}_i := \bot$
2: **repeat**
3: $\quad \overrightarrow{X}_i := \overrightarrow{X}_{i-1} \nabla_T F(\overrightarrow{X}_{i-1}) \Longrightarrow \{50, 10, 0\}$.
4: **until** $\overrightarrow{X}_i \sqsubseteq \overrightarrow{X}_{i-1}$

Exemple

1: x = [90,100];
2: While $(x \geq 0)$
3:{
4: y := $f(x)$/x;
5: x := x-1;
6: }
7:

$\Longrightarrow$

Calcul d'un point-fixe

Itération:4

	x
1	[90,100]
2	[0,100]
5	[9,99]
7	$\bot$

Le widening avec seuils

Algorithm 20 Méthode d'itération de Kleene avec widening avec seuils

1: $\overrightarrow{X}_i := \bot$
2: **repeat**
3: $\quad \overrightarrow{X}_i := \overrightarrow{X}_{i-1} \nabla_T F(\overrightarrow{X}_{i-1}) \Longrightarrow \{50, 10, 0\}$.
4: **until** $\overrightarrow{X}_i \sqsubseteq \overrightarrow{X}_{i-1}$

Exemple

1: x = [90,100];
2: While $(x \geq 0)$
3:{
4: y := $f(x)$/x;$\Longrightarrow$ Error
5: x := x-1;
6: }
7:

$\Longrightarrow$

Calcul d'un point-fixe

Itération:5

	x
1	[90,100]
2	[0,100]
5	[-1,99]
7	$\bot$

Le widening avec seuils

Algorithm 21 Méthode d'itération de Kleene avec widening avec seuils

1: $\overrightarrow{X}_i := \bot$
2: **repeat**
3: $\quad \overrightarrow{X}_i := \overrightarrow{X}_{i-1} \nabla_T F(\overrightarrow{X}_{i-1}) \Longrightarrow \{50, 10, 0\}$.
4: **until** $\overrightarrow{X}_i \sqsubseteq \overrightarrow{X}_{i-1}$

Exemple

1: x = [90,100];
2: While $(x \geq 0)$
3:{
4: y := $f(x)$/x;$\Longrightarrow$ Error
5: x := x-1;
6: }
7:

$\Longrightarrow$

Calcul d'un point-fixe

Itération:5

	x
1	[90,100]
2	[-1,100]
5	[-1,99]
7	$\bot$

Le widening avec seuils

Algorithm 22 Méthode d'itération de Kleene avec widening avec seuils

1: $\overrightarrow{X}_i := \bot$
2: **repeat**
3: $\quad \overrightarrow{X}_i := \overrightarrow{X}_{i-1} \nabla_T F(\overrightarrow{X}_{i-1}) \Longrightarrow \{50, 10, 0\}$.
4: **until** $\overrightarrow{X}_i \sqsubseteq \overrightarrow{X}_{i-1}$

Exemple

1: x = [90,100];
2: While $(x \geq 0)$
3:{
4: y := $f(x)$/x;$\Longrightarrow$ Error
5: x := x-1;
6: }
7:

$\Longrightarrow$

Calcul d'un point-fixe

Itération: 6

	x
1	[90,100]
2	[-1,100]
5	[-1,99]
7	[-1,-1]

Le widening avec seuils

Algorithm 23 Méthode d'itération de Kleene avec widening avec seuils

1: $\overrightarrow{X}_i := \bot$
2: **repeat**
3: $\quad \overrightarrow{X}_i := \overrightarrow{X}_{i-1} \nabla_T F(\overrightarrow{X}_{i-1})$
4: **until** $\overrightarrow{X}_i \sqsubseteq \overrightarrow{X}_{i-1}$

- Avantages :
  - Permet un calcul rapide.
  - Gagner de la précision.
- Inconvénients :
  - L'ensemble des seuils doit être déterminé statiquement par le programmeur.

Le widening avec seuils

Algorithm 24 Méthode d'itération de Kleene avec widening avec seuils

1: $\overrightarrow{X}_i := \bot$
2: **repeat**
3: $\quad \overrightarrow{X}_i := \overrightarrow{X}_{i-1} \nabla_T F(\overrightarrow{X}_{i-1})$
4: **until** $\overrightarrow{X}_i \sqsubseteq \overrightarrow{X}_{i-1}$

- Avantages :
  - Permet un calcul rapide.
  - Gagner de la précision.
- Inconvénients :
  - L'ensemble des seuils doit être déterminé statiquement par le programmeur.

Notre approche consiste à trouver dynamiquement des seuils pertinents.

Plan

Accélération de calcul du point-fixe en interprétation abstraite en utilisant la convergence des suites numériques.

1. Analyse statique par interprétation abstraite

2. Calcul du point-fixe

3. Méthodes d'accélération de convergence en analyse numérique
   - Accélération de convergence des suites numériques
   - La méthode Δ^2-d'Aitken
   - ε-Algorithme
   - ε-Algorithme Vectoriel

4. Accélération de calcul du point-fixe

Accélération de convergence des suites numériques

Une suite convergente

Soit (S_n) une suite numérique et S un réel. Nous disons que (S_n) converge vers S ssi $\lim_{n \to +\infty} S_n = S$.

Accélération de convergence

Soit (S_n) et (Y_n) deux suites numériques qui admettent un réel S comme limite. Nous disons que (Y_n) converge vers S plus rapidement que (S_n) si :

$$\lim_{n \to \infty} \frac{Y_n - S}{S_n - S} = 0$$

Accélération de convergence des suites numériques

Le procédé de transformation de suites

Un procédé de transformation de suites est une fonction P qui associe à toute suite numérique (S_n) une nouvelle suite (Y_n) telle que, si (S_n) converge vers S, alors (Y_n) converge plus rapidement vers S.

Le noyau d'un procédé

Soit P un procédé de transformation. Le noyau de P, noté θ, représente l'ensemble des suites y, que ce procédé transforme en la suite constante S.

$$\theta = \{(\exists S \in \mathbb{R}), y | P(y) = S\}. \tag{1}$$

Accélération de convergence des suites numériques

Le procédé de transformation de suites

Un procédé de transformation de suites est une fonction P qui associe à toute suite numérique (S_n) une nouvelle suite (Y_n) telle que, si (S_n) converge vers S, alors (Y_n) converge plus rapidement vers S.

Le noyau d'un procédé

Soit P un procédé de transformation. Le noyau de P, noté θ, représente l'ensemble des suites y, que ce procédé transforme en la suite constante S.

$$\theta = \{(\exists S \in \mathbb{R}), y | P(y) = S\}. \quad (1)$$

Il existe plusieurs méthodes de transformations.

La méthode Δ^2-d'Aitken

Définition

Soit (S_n) la suite initiale et (S'_n) la suite transformée telle que:

$$\forall n \in \mathbb{N}, S'_{n+1} = \frac{S_{n+1} - S_n}{S_{n+2} - 2S_{n+1} + S_n}.$$

La méthode Δ^2-d'Aitken

Définition

Soit (S_n) la suite initiale et (S'_n) la suite transformée telle que:
$\forall n \in \mathbb{N}, S'_{n+1} = \dfrac{S_{n+1} - S_n}{S_{n+2} - 2S_{n+1} + S_n}$.

Proposition

Le noyau du procédé d'Aitken contient les suites exponentiellement convergentes qui s'ecrivent sous la forme: $\forall n \in \mathbb{N}, S_n = S + \alpha\lambda^n$. (avec $S \in \mathbb{R}$.)

Exemple

Exemple

Prenons la suite: $S_n = 1 + \frac{1}{n+1}, \forall n \geq \mathbb{N}$ avec $\lim_{n \to +\infty} S_n = 1$

Exemple

Exemple

Prenons la suite: $S_n = 1 + \frac{1}{n+1}, \forall n \geq \mathbb{N}$ avec $\lim_{n \to +\infty} S_n = 1$

Résultats

ε_n^0	ε_n^2	ε_n^4	ε_n^6	ε_n^8
2.0000000				
1.5000000				
1.3333333				
1.2500000				
1.2000000				
1.1666667				
1.1428571				
1.1250000				
1,1111111				
1,1000000				

Exemple

Exemple

Prenons la suite: $S_n = 1 + \frac{1}{n+1}, \forall n \geq \mathbb{N}$ avec $\lim_{n \to +\infty} S_n = 1$

Résultats

ε_n^0	ε_n^2	ε_n^4	ε_n^6	ε_n^8
2.0000000				
1.5000000	1.2500000			
1.3333333				
1.2500000				
1.2000000				
1.1666667				
1.1428571				
1.1250000				
1,1111111				
1,1000000				

Exemple

Exemple

Prenons la suite: $S_n = 1 + \frac{1}{n+1}, \forall n \geq \mathbb{N}$ avec $\lim_{n \to +\infty} S_n = 1$

Résultats

ε_n^0	ε_n^2	ε_n^4	ε_n^6	ε_n^8
2.0000000				
1.5000000	1.2500000			
1.3333333	1.1666667			
1.2500000				
1.2000000				
1.1666667				
1.1428571				
1.1250000				
1,1111111				
1,1000000				

Exemple

Exemple

Prenons la suite: $S_n = 1 + \frac{1}{n+1}, \forall n \geq \mathbb{N}$ avec $\lim_{n \to +\infty} S_n = 1$

Résultats

ε_n^0	ε_n^2	ε_n^4	ε_n^6	ε_n^8
2.0000000				
1.5000000	1.2500000			
1.3333333	1.1666667			
1.2500000	1.1249999			
1.2000000				
1.1666667				
1.1428571				
1.1250000				
1,1111111				
1,1000000				

Exemple

Exemple

Prenons la suite: $S_n = 1 + \frac{1}{n+1}, \forall n \geq \mathbb{N}$ avec $\lim_{n \to +\infty} S_n = 1$

Résultats

ε_n^0	ε_n^2	ε_n^4	ε_n^6	ε_n^8
2.0000000				
1.5000000	1.2500000			
1.3333333	1.1666667			
1.2500000	1.1249999			
1.2000000	1.1000001			
1.1666667				
1.1428571				
1.1250000				
1,1111111				
1,1000000				

Exemple

Exemple

Prenons la suite: $S_n = 1 + \frac{1}{n+1}, \forall n \geq \mathbb{N}$ avec $\lim_{n \to +\infty} S_n = 1$

Résultats

ε_n^0	ε_n^2	ε_n^4	ε_n^6	ε_n^8
2.0000000				
1.5000000	1.2500000			
1.3333333	1.1666667			
1.2500000	1.1249999			
1.2000000	1.1000001			
1.1666667	1.0833333			
1.1428571				
1.1250000				
1,1111111				
1,1000000				

Exemple

Exemple

Prenons la suite: $S_n = 1 + \frac{1}{n+1}, \forall n \geq \mathbb{N}$ avec $\lim_{n \to +\infty} S_n = 1$

Résultats

ε_n^0	ε_n^2	ε_n^4	ε_n^6	ε_n^8
2.0000000				
1.5000000	1.2500000			
1.3333333	1.1666667			
1.2500000	1.1249999			
1.2000000	1.1000001			
1.1666667	1.0833333			
1.1428571	1.0714287			
1.1250000				
1,1111111				
1,1000000				

Exemple

Exemple

Prenons la suite: $S_n = 1 + \frac{1}{n+1}, \forall n \geq \mathbb{N}$ avec $\lim_{n \to +\infty} S_n = 1$

Résultats

ε_n^0	ε_n^2	ε_n^4	ε_n^6	ε_n^8
2.0000000				
1.5000000	1.2500000			
1.3333333	1.1666667			
1.2500000	1.1249999			
1.2000000	1.1000001			
1.1666667	1.0833333			
1.1428571	1.0714287			
1.1250000	1.0624998			
1,1111111				
1,1000000				

Exemple

Exemple

Prenons la suite: $S_n = 1 + \frac{1}{n+1}, \forall n \geq \mathbb{N}$ avec $\lim_{n \to +\infty} S_n = 1$

Résultats

ε_n^0	ε_n^2	ε_n^4	ε_n^6	ε_n^8
2.0000000				
1.5000000	1.2500000			
1.3333333	1.1666667			
1.2500000	1.1249999			
1.2000000	1.1000001			
1.1666667	1.0833333			
1.1428571	1.0714287			
1.1250000	1.0624998			
1,1111111	1.0555557			
1,1000000				

Exemple

Exemple

Prenons la suite: $S_n = 1 + \frac{1}{n+1}, \forall n \geq \mathbb{N}$ avec $\lim_{n \to +\infty} S_n = 1$

Résultats

ε_n^0	ε_n^2	ε_n^4	ε_n^6	ε_n^8
2.0000000				
1.5000000	1.2500000			
1.3333333	1.1666667			
1.2500000	1.1249999			
1.2000000	1.1000001			
1.1666667	1.0833333			
1.1428571	1.0714287			
1.1250000	1.0624998			
1,1111111	1.0555557			
1,1000000				

Exemple

Exemple

Prenons la suite: $S_n = 1 + \frac{1}{n+1}, \forall n \geq \mathbb{N}$ avec $\lim_{n \to +\infty} S_n = 1$

Résultats

ε_n^0	ε_n^2	ε_n^4	ε_n^6	ε_n^8
2.0000000				
1.5000000	1.2500000			
1.3333333	1.1666667	1.1111109		
1.2500000	1.1249999	1.0833337		
1.2000000	1.1000001	1.0666663		
1.1666667	1.0833333	1.0555556		
1.1428571	1.0714287	1.0476161		
1.1250000	1.0624998	1.0416761		
1,1111111	1.0555557			
1,1000000				

Exemple

Exemple

Prenons la suite: $S_n = 1 + \frac{1}{n+1}, \forall n \geq \mathbb{N}$ avec $\lim_{n \to +\infty} S_n = 1$

Résultats

ε_n^0	ε_n^2	ε_n^4	ε_n^6	ε_n^8
2.0000000				
1.5000000	1.2500000			
1.3333333	1.1666667	1.1111109		
1.2500000	1.1249999	1.0833337	1.0624931	
1.2000000	1.1000001	1.0666663	1.0500028	
1.1666667	1.0833333	1.0555556	1.0416545	
1.1428571	1.0714287	1.0476161	1.0357504	
1.1250000	1.0624998	1.0416761		
1,1111111	1.0555557			
1,1000000				

Exemple

Exemple

Prenons la suite: $S_n = 1 + \frac{1}{n+1}, \forall n \geq \mathbb{N}$ avec $\lim_{n \to +\infty} S_n = 1$

Résultats

ε_n^0	ε_n^2	ε_n^4	ε_n^6	ε_n^8
2.0000000				
1.5000000	1.2500000			
1.3333333	1.1666667	1.1111109		
1.2500000	1.1249999	1.0833337	1.0624931	
1.2000000	1.1000001	1.0666663	1.0500028	1.0399799
1.1666667	1.0833333	1.0555556	1.0416545	1.0334257
1.1428571	1.0714287	1.0476161	1.0357504	
1.1250000	1.0624998	1.0416761		
1,1111111	1.0555557			
1,1000000				

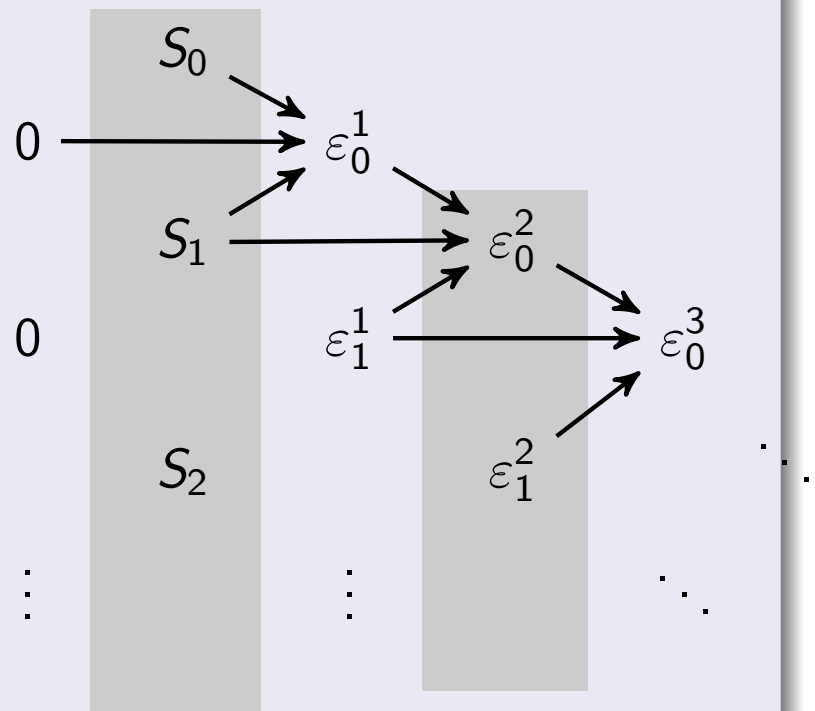
ε-Algorithme [Wynn/Shanks/Brezinski]

Définition

Soit la suite $(S_n), (\forall k > 0, \forall n \in \mathbb{N})$:

$$\begin{cases} \epsilon_n^{-1} = 0 \\ \epsilon_n^0 = S_n \\ \epsilon_n^{k+1} = \epsilon_{n+1}^{k-1} + \dfrac{1}{\epsilon_{n+1}^k - \epsilon_n^k} \end{cases}$$

ε-tableau



- Accélération des suites de colonnes paires.
- Accélération des suites en diagonales des éléments de colonnes paires.

ε-Algorithme [Brezinski77]

Proposition 01

Le noyau de l'ε-Algorithme est assez important, il permet d'accélérer quelques suites monotones et oscillantes, ainsi que toutes les suites du types $X_{n+1} = AX_n + B$.

Proposition 02

Soient (S_n) une suite convergente et $(S'_n) = (\varepsilon_0^{2k})$ sa suite accélérée par l'ε-Algorithme. Calculer p éléments de (S'_n) nécessite $(2p-1)$ éléments de (S_n).

ε-Algorithme Vectoriel

Définition

Soit la suite $(S_n), (\forall k > 0, \forall n \in \mathbb{N})$:

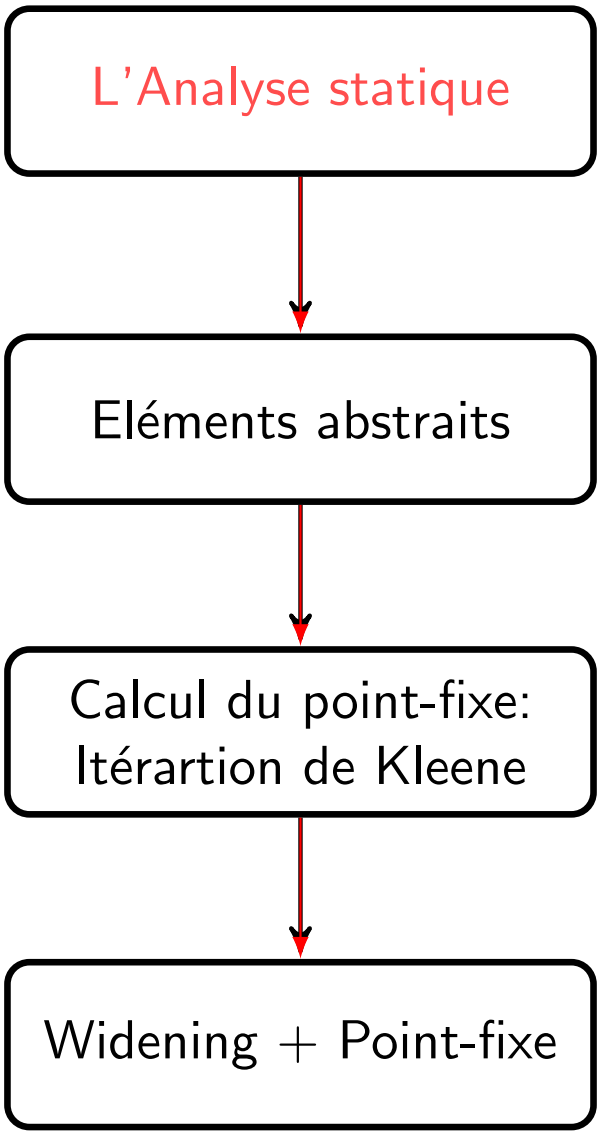
$$\begin{cases} \epsilon_n^{-1} = 0 \\ \epsilon_n^0 = S_n \\ \epsilon_n^{k+1} = \epsilon_{n+1}^{k-1} + \dfrac{1}{(\epsilon_{n+1}^k - \epsilon_n^k)^{-1}} \end{cases}$$

L'inverse d'un vecteur d'aprés Brezinski

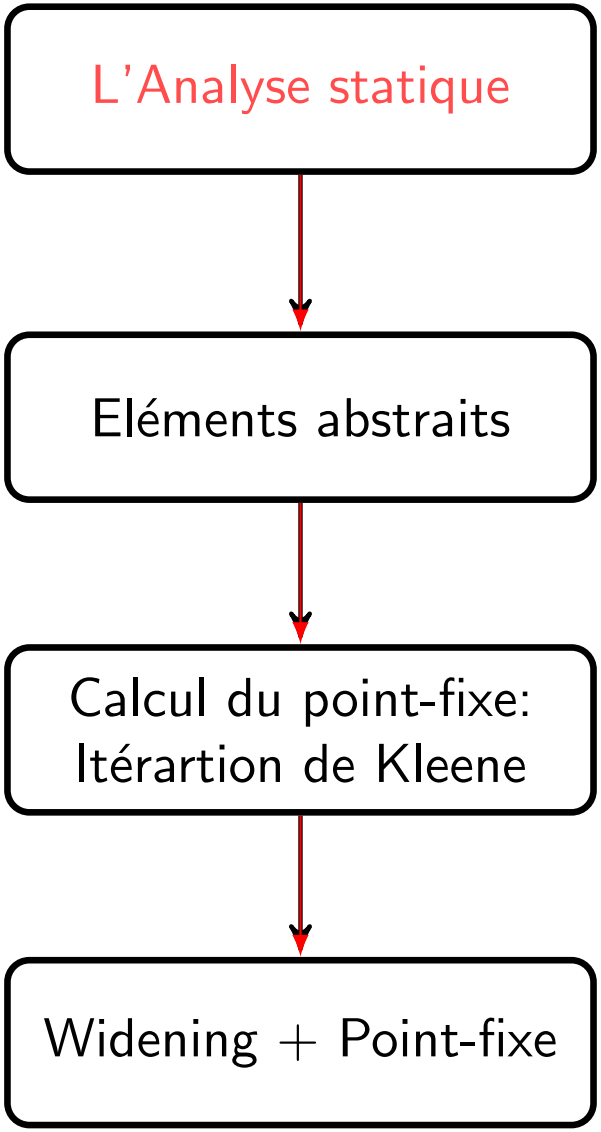
Soit V un vecteur à p composantes, avec $p \in \mathbb{N}$. Et soit V^{-1} le vecteur inverse de V. Nous définissons V^{-1} par $V^{-1} = \dfrac{V}{V \odot V}$ avec $\odot$ le produit scalaire de deux vecteurs.

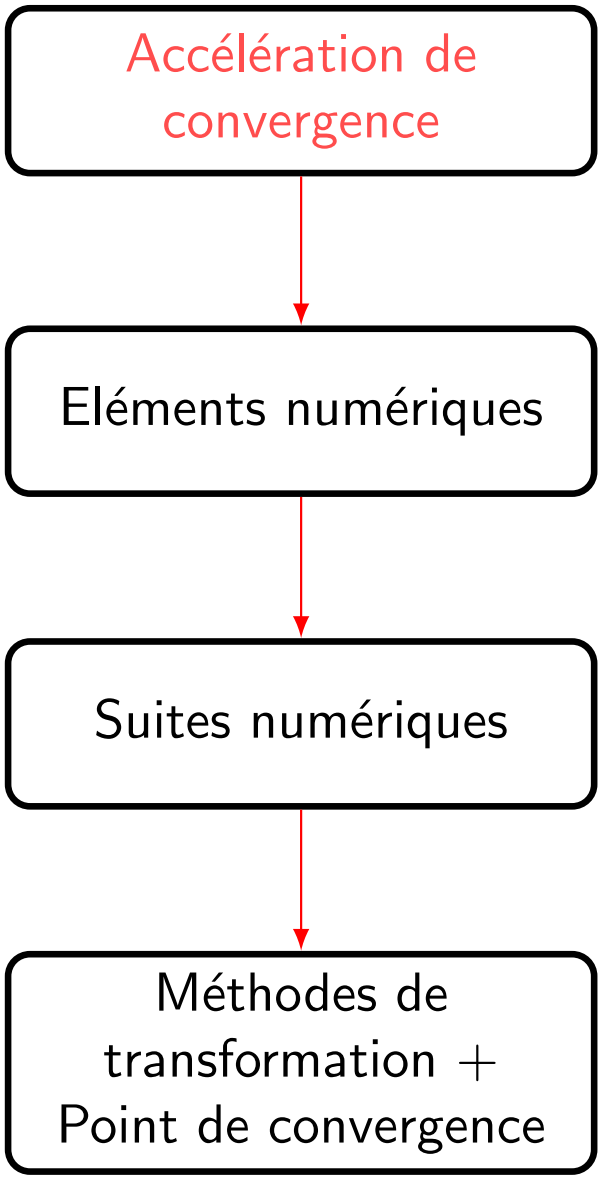
- Le noyau de l'ε-Algorithme Vectoriel contient les suites vectorielles de la forme $V_{n+1} = AV_n + B$ tel que A est une matrice et B un vecteur.

Récapitulatif

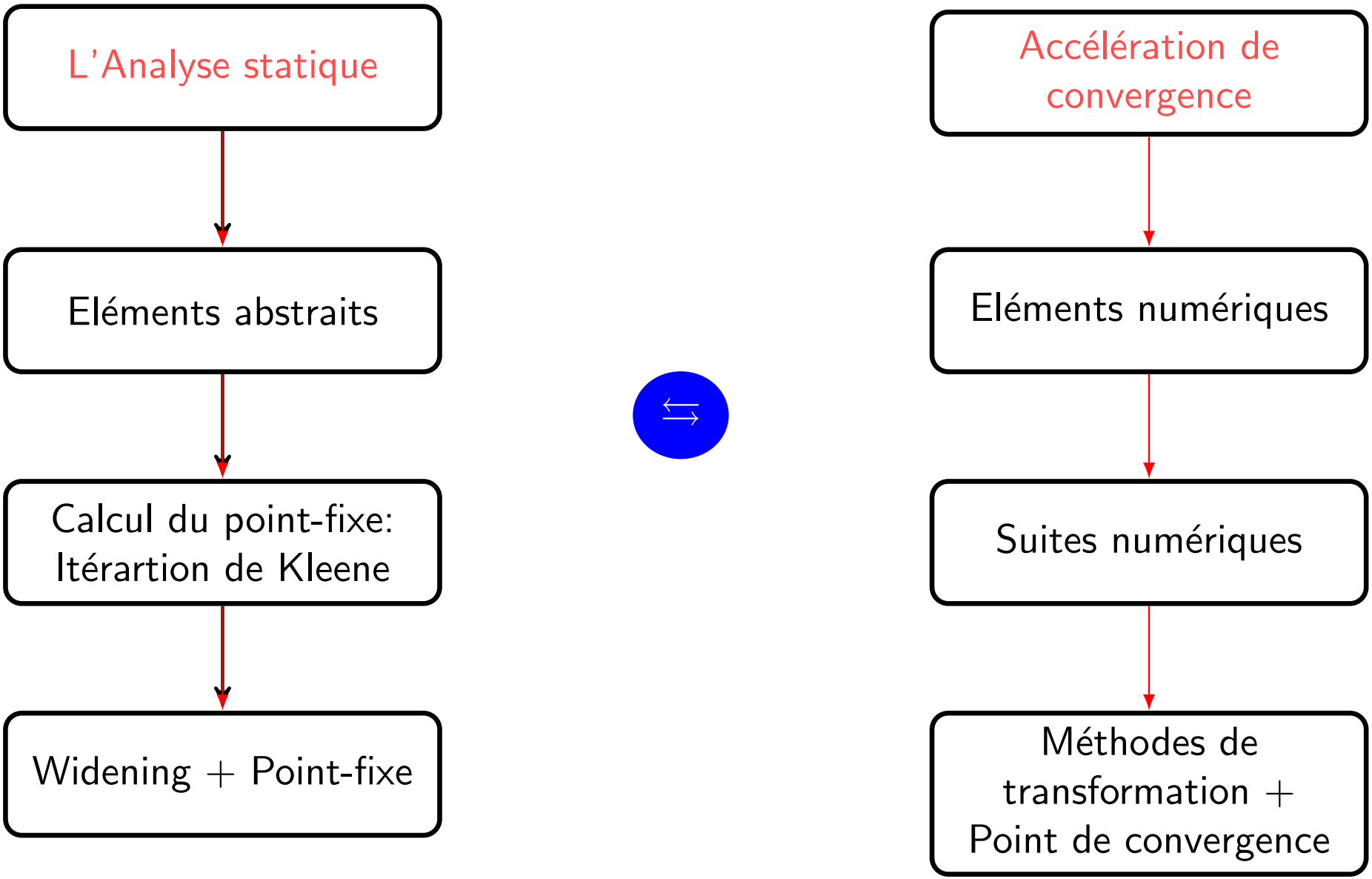


Récapitulatif





Récapitulatif



Plan

Accélération de calcul du point-fixe en interprétation abstraite en utilisant la convergence des suites numériques.

Méthodologie par l'exemple

Le programme

```
while (1) {*

xn1 = -0.4375*x1+0.0625*x2+0.2652*x3+0.1*u1;

xn2 = 0.0625*x1+0.4375*x2+0.2652*x3+0.1*u2;

xn3 = -0.2652*x1+0.2652*x2+0.375*x3+0.1*u3;

x1 = xn1;

x2 = xn2;

x3 = xn3;

}
```

$\Longrightarrow$

Chaque itération

x_1

$[1.000000, 2.000000]$

Méthodologie par l'exemple

Le programme

```
while (1) {*
xn1 = -0.4375*x1+0.0625*x2+0.2652*x3+0.1*u1;
xn2 = 0.0625*x1+0.4375*x2+0.2652*x3+0.1*u2;
xn3 = -0.2652*x1+0.2652*x2+0.375*x3+0.1*u3;
x1 = xn1;
x2 = xn2;
x3 = xn3;
}
```

$\Longrightarrow$

Chaque itération

x_1

$[1.000000, 2.000000]$
$[-0.447300, 5.716500]$

Méthodologie par l'exemple

Le programme

```
while (1) {*

xn1 = -0.4375*x1+0.0625*x2+0.2652*x3+0.1*u1;

xn2 = 0.0625*x1+0.4375*x2+0.2652*x3+0.1*u2;

xn3 = -0.2652*x1+0.2652*x2+0.375*x3+0.1*u3;

x1 = xn1;

x2 = xn2;

x3 = xn3;

}
```

$\Longrightarrow$

Chaque itération

x_1

$[1.000000, 2.000000]$
$[-0.447300, 5.716500]$
$[-2.291255, 6.573381]$

Méthodologie par l'exemple

Le programme

```
while (1) {*
xn1 = -0.4375*x1+0.0625*x2+0.2652*x3+0.1*u1;
xn2 = 0.0625*x1+0.4375*x2+0.2652*x3+0.1*u2;
xn3 = -0.2652*x1+0.2652*x2+0.375*x3+0.1*u3;
x1 = xn1;
x2 = xn2;
x3 = xn3;
}
```

$\Longrightarrow$

Chaque itération

x_1

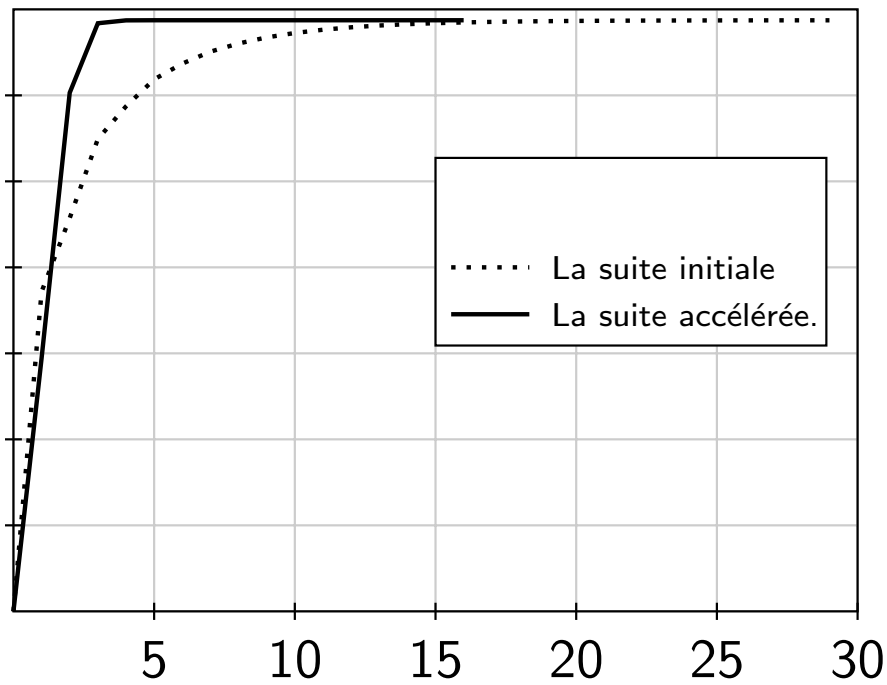
$[1.000000, 2.000000]$
$[-0.447300, 5.716500]$
$[-2.291255, 6.573381]$

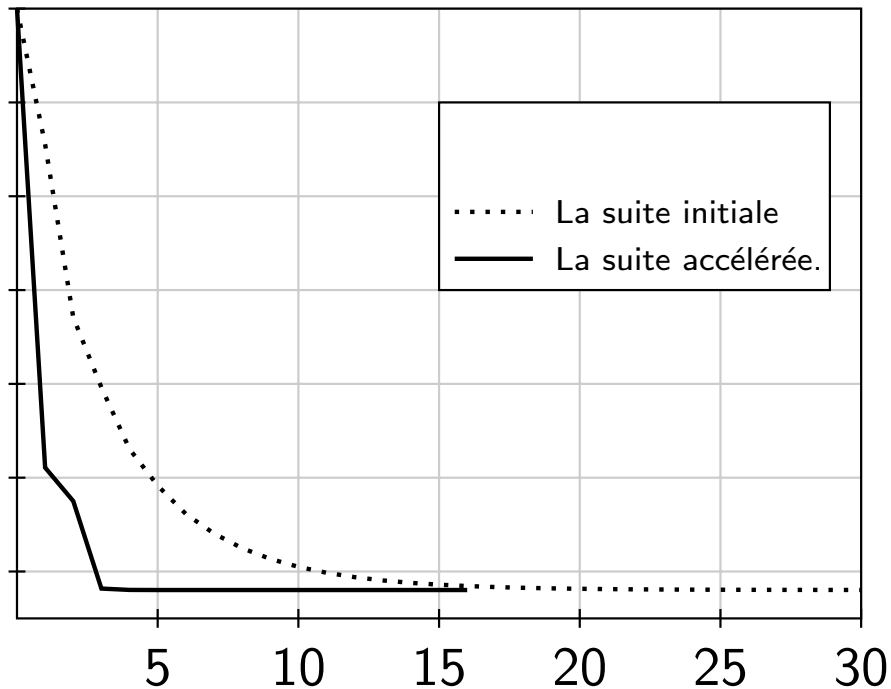
$\Downarrow$ 127 itérations

$[-5.197505, 8.873306]$

Méthodologie par l'exemple

- Construction des suites de bornes supérieures et inférieures à partir des suites d'intervalles.





- Reconsruction d'un intevalle à partir des résultats obtenus.

Algorithme de Kleene accéléré

Algorithm 25 L'itération de Kleene accélérée

1: **repeat**
2: $\quad \overrightarrow{X_i} := \overrightarrow{X_{i-1}} \sqcup F(\overrightarrow{X_{i-1}})$
3: $\quad \overrightarrow{y_i} :=$ Accelerate $\left(\Lambda_A(\overrightarrow{X_0}), \ldots, \Lambda_A(\overrightarrow{X_i})\right)$
4: $\quad$ **if** $||\overrightarrow{y_i} - \overrightarrow{y_{i-1}}|| \leq \delta$ **then**
5: $\quad\quad \overrightarrow{X_i} := \overrightarrow{X_i} \sqcup \Upsilon_A(\overrightarrow{y_i})$
6: $\quad$ **end if**
7: **until** $\overrightarrow{X_i} \sqsubseteq \overrightarrow{X_{i-1}}$

Algorithme de Kleene accéléré

- Application des méthodes d'accélération.

Algorithm 26 L'itération de Kleene accélérée

1: **repeat**
2: $\overrightarrow{X_i} := \overrightarrow{X_{i-1}} \sqcup F(\overrightarrow{X_{i-1}})$
3: $\overrightarrow{y_i} :=$ Accelerate$(\Lambda_A(X_0), \ldots, \Lambda_A(\overrightarrow{X_i}))$
4: **if** $||\overrightarrow{y_i} - \overrightarrow{y_{i-1}}|| \leq \delta$ **then**
5: $\overrightarrow{X_i} := \overrightarrow{X_i} \sqcup \Upsilon_A(\overrightarrow{y_i})$
6: **end if**
7: **until** $\overrightarrow{X_i} \sqsubseteq \overrightarrow{X_{i-1}}$

Algorithme de Kleene accéléré

- Application des méthodes d'accélération.

Algorithm 27 L'itération de Kleene accélérée

1: **repeat**
2: $\overrightarrow{X_i} := \overrightarrow{X_{i-1}} \sqcup F(\overrightarrow{X_{i-1}})$
3: $\overrightarrow{y_i} :=$ Accelerate$(\Lambda_A(X_0), \ldots, \Lambda_A(\overrightarrow{X_i}))$
4: **if** $||\overrightarrow{y_i} - \overrightarrow{y_{i-1}}|| \leq \delta$ **then**
5: $\overrightarrow{X_i} := \overrightarrow{X_i} \sqcup \Upsilon_A(\overrightarrow{y_i})$
6: **end if**
7: **until** $\overrightarrow{X_i} \sqsubseteq \overrightarrow{X_{i-1}}$

- La fonction *Extraction*

Algorithme de Kleene accéléré

- Application des méthodes d'accélération.

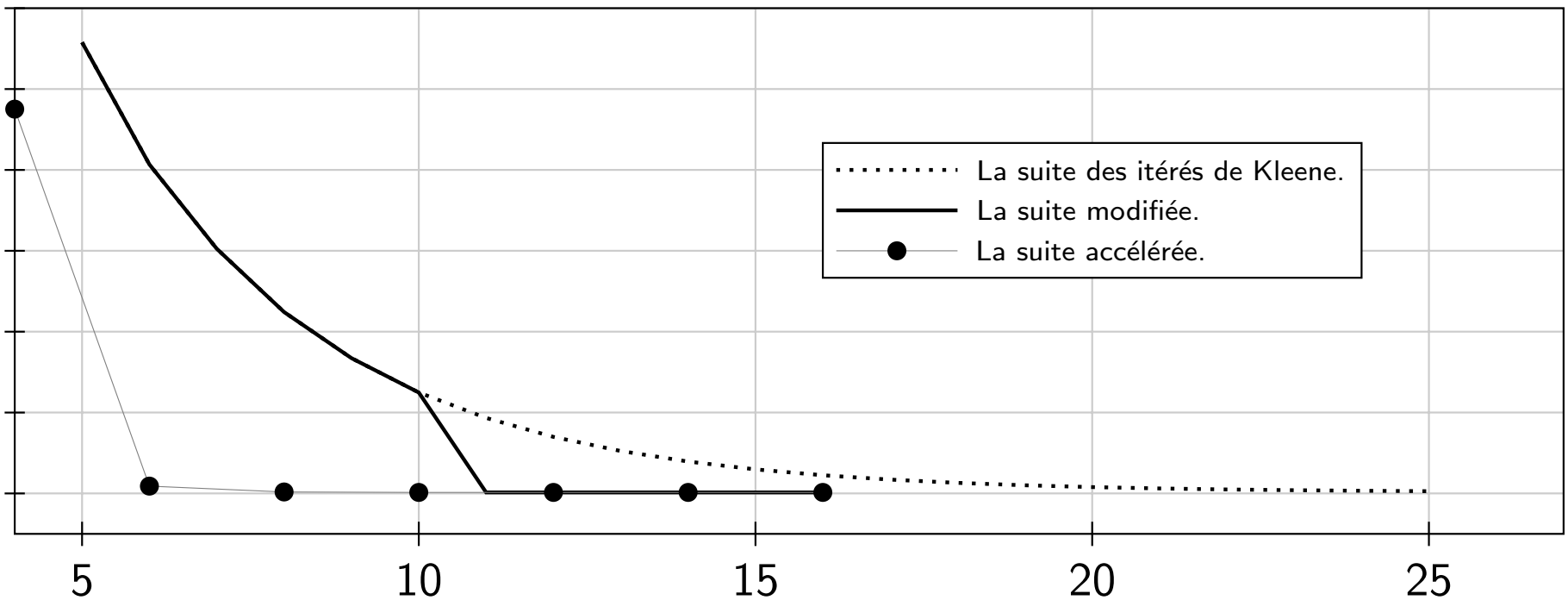
Algorithm 28 L'itération de Kleene accélérée

1: **repeat**
2: $\overrightarrow{X_i} := \overrightarrow{X_{i-1}} \sqcup F(\overrightarrow{X_{i-1}})$
3: $\overrightarrow{y_i} :=$ Accelerate$(\Lambda_A(X_0), \ldots, \Lambda_A(\overrightarrow{X_i}))$
4: **if** $||\overrightarrow{y_i} - \overrightarrow{y_{i-1}}|| \leq \delta$ **then**
5: $\overrightarrow{X_i} := \overrightarrow{X_i} \sqcup \Upsilon_A(\overrightarrow{y_i})$
6: **end if**
7: **until** $\overrightarrow{X_i} \sqsubseteq \overrightarrow{X_{i-1}}$

- La fonction *Extraction*
- La fonction *Combination*

Application de l'algorithme de Kleene accéléré

- Méthode d'accélération du calcul du point-fixe appliquée à la suite des bornes inférieures de x_1.



Méthodologie par l'exemple

```
while (1) {
xn1 = -0.4375 * x1 + 0.0625 * x2 + 0.2652 * x3 + 0.1 * u1;
xn2 = 0.0625 * x1 + 0.4375 * x2 + 0.2652 * x3 + 0.1 * u2;
xn3 = -0.2652 * x1 + 0.2652 * x2 + 0.375 * x3 + 0.1 * u3;
x1 = xn1;x2 = xn2;x3 = xn3;
} [-5.197505,8.873306]
```

Itération:3

x	$\Upsilon_A(\overrightarrow{y})$	$\|y_i - y_{i-1}\|$	**Seuils** ($\delta = 10^{-3}$)
[1.000000,2.000000]	[1.000000,2.000000]		
[-0.447300,5.716500]			
[-2.291255,6.573381]			

Méthodologie par l'exemple

```
while (1) {
xn1 = -0.4375 * x1 + 0.0625 * x2 + 0.2652 * x3 + 0.1 * u1;
xn2 = 0.0625 * x1 + 0.4375 * x2 + 0.2652 * x3 + 0.1 * u2;
xn3 = -0.2652 * x1 + 0.2652 * x2 + 0.375 * x3 + 0.1 * u3;
x1 = xn1;x2 = xn2;x3 = xn3;
} [-5.197505,8.873306]
```

Itération:3

x	$\Upsilon_A(\overrightarrow{y})$	$\|y_i - y_{i-1}\|$	**Seuils ($\delta = 10^{-3}$)**
[1.000000,2.000000]	[1.000000,2.000000]		
[-0.447300,5.716500]			
[-2.291255,6.573381]	[6.280857,6.830145]	max(5.280857,4.830145)	

Méthodologie par l'exemple

```
while (1) {
xn1 = -0.4375 * x1 + 0.0625 * x2 + 0.2652 * x3 + 0.1 * u1;
xn2 = 0.0625 * x1 + 0.4375 * x2 + 0.2652 * x3 + 0.1 * u2;
xn3 = -0.2652 * x1 + 0.2652 * x2 + 0.375 * x3 + 0.1 * u3;
x1 = xn1;x2 = xn2;x3 = xn3;
} [-5.197505,8.873306]
```

Itération:5

x	$\Upsilon_A(\overrightarrow{y})$	$\|y_i - y_{i-1}\|$	**Seuils** ($\delta = 10^{-3}$)
[1.000000,2.000000]	[1.000000,2.000000]		
[-0.447300,5.716500]			
[-2.291255,6.573381]	[6.280857,6.830145]	max(5.280857,4.830145)	
[-3.038029,7.492492]			
[-3.695558,7.871720]	[-4.663282,8.463092]	max(10.944139,2.367053)	

Méthodologie par l'exemple

```
while (1) {
xn1 = -0.4375 * x1 + 0.0625 * x2 + 0.2652 * x3 + 0.1 * u1;
xn2 = 0.0625 * x1 + 0.4375 * x2 + 0.2652 * x3 + 0.1 * u2;
xn3 = -0.2652 * x1 + 0.2652 * x2 + 0.375 * x3 + 0.1 * u3;
x1 = xn1;x2 = xn2;x3 = xn3;
} [-5.197505,8.873306]
```

Itération:7

x	$\Upsilon_A(\overrightarrow{y})$	$\|y_i - y_{i-1}\|$	**Seuils** ($\delta = 10^{-3}$)
[1.000000,2.000000]	[1.000000,2.000000]		
[-0.447300,5.716500]			
[-2.291255,6.573381]	[6.280857,6.830145]	max(5.280857,4.830145)	
[-3.038029,7.492492]			
[-3.695558,7.871720]	[-4.663282,8.463092]	max(10.944139,2.367053)	
[-4.084503,8.185954]			
[-4.386442,8.369221]	[-5.189239,8.851176]	max(0.525957,0.388084)	

Méthodologie par l'exemple

```
while (1) {
xn1 = -0.4375 * x1 + 0.0625 * x2 + 0.2652 * x3 + 0.1 * u1;
xn2 = 0.0625 * x1 + 0.4375 * x2 + 0.2652 * x3 + 0.1 * u2;
xn3 = -0.2652 * x1 + 0.2652 * x2 + 0.375 * x3 + 0.1 * u3;
x1 = xn1;x2 = xn2;x3 = xn3;
} [-5.197505,8.873306]
```

Itération:9

x	$\Upsilon_A(\overrightarrow{y})$	$\|y_i - y_{i-1}\|$	**Seuils** ($\delta = 10^{-3}$)
[1.000000,2.000000]	[1.000000,2.000000]		
[-0.447300,5.716500]			
[-2.291255,6.573381]	[6.280857,6.830145]	max(5.280857,4.830145)	
[-3.038029,7.492492]			
[-3.695558,7.871720]	[-4.663282,8.463092]	max(10.944139,2.367053)	
[-4.084503,8.185954]			
[-4.386442,8.369221]	[-5.189239,8.851176]	max(0.525957,0.388084)	
[-4.594886,8.508279]			
[-4.751445,8.603235]	[-5.197089,8.872567]	max(0.00785,0.021391)	

Méthodologie par l'exemple

```
while (1) {
xn1 = -0.4375 * x1 + 0.0625 * x2 + 0.2652 * x3 + 0.1 * u1;
xn2 = 0.0625 * x1 + 0.4375 * x2 + 0.2652 * x3 + 0.1 * u2;
xn3 = -0.2652 * x1 + 0.2652 * x2 + 0.375 * x3 + 0.1 * u3;
x1 = xn1;x2 = xn2;x3 = xn3;
} [-5.197505,8.873306]
```

Itération:11

x	$\Upsilon_A(\overrightarrow{y})$	$\|y_i - y_{i-1}\|$	**Seuils** ($\delta = 10^{-3}$)
[1.000000,2.000000]	[1.000000,2.000000]		
[-0.447300,5.716500]			
[-2.291255,6.573381]	[6.280857,6.830145]	max(5.280857,4.830145)	
[-3.038029,7.492492]			
[-3.695558,7.871720]	[-4.663282,8.463092]	max(10.944139,2.367053)	
[-4.084503,8.185954]			
[-4.386442,8.369221]	[-5.189239,8.851176]	max(0.525957,0.388084)	
[-4.594886,8.508279]			
[-4.751445,8.603235]	[-5.197089,8.872567]	max(0.00785,0.021391)	
[-4.865385,8.673918]			
[-4.950393,8.725095]	[-5.197506,8.873307]	max(0.000417,0.00074)	[-5.197506,8.873307]

Méthodologie par l'exemple

```
while (1) {
xn1 = -0.4375 * x1 + 0.0625 * x2 + 0.2652 * x3 + 0.1 * u1;
xn2 = 0.0625 * x1 + 0.4375 * x2 + 0.2652 * x3 + 0.1 * u2;
xn3 = -0.2652 * x1 + 0.2652 * x2 + 0.375 * x3 + 0.1 * u3;
x1 = xn1;x2 = xn2;x3 = xn3;
} [-5.197505,8.873306]
```

Itération: 12

x	$\Upsilon_A(\overrightarrow{y})$	$\|y_i - y_{i-1}\|$	**Seuils** ($\delta = 10^{-3}$)
[1.000000,2.000000]	[1.000000,2.000000]		
[-0.447300,5.716500]			
[-2.291255,6.573381]	[6.280857,6.830145]	max(5.280857,4.830145)	
[-3.038029,7.492492]			
[-3.695558,7.871720]	[-4.663282,8.463092]	max(10.944139,2.367053)	
[-4.084503,8.185954]			
[-4.386442,8.369221]	[-5.189239,8.851176]	max(0.525957,0.388084)	
[-4.594886,8.508279]			
[-4.751445,8.603235]	[-5.197089,8.872567]	max(0.00785,0.021391)	
[-4.865385,8.673918]			
[-4.950393,8.725095]	[-5.197506,8.873307]	max(0.000417,0.00074)	[-5.197506,8.873307]
[-5.197506,8.873307]			

Méthodologie par l'exemple

```
while (1) {
xn1 = -0.4375 * x1 + 0.0625 * x2 + 0.2652 * x3 + 0.1 * u1;
xn2 = 0.0625 * x1 + 0.4375 * x2 + 0.2652 * x3 + 0.1 * u2;
xn3 = -0.2652 * x1 + 0.2652 * x2 + 0.375 * x3 + 0.1 * u3;
x1 = xn1;x2 = xn2;x3 = xn3;
} [-5.197505,8.873306]
```

Itération: 13

x	$\Upsilon_A(\overrightarrow{y})$	$\|y_i - y_{i-1}\|$	**Seuils** ($\delta = 10^{-3}$)
[1.000000,2.000000]	[1.000000,2.000000]		
[-0.447300,5.716500]			
[-2.291255,6.573381]	[6.280857,6.830145]	max(5.280857,4.830145)	
[-3.038029,7.492492]			
[-3.695558,7.871720]	[-4.663282,8.463092]	max(10.944139,2.367053)	
[-4.084503,8.185954]			
[-4.386442,8.369221]	[-5.189239,8.851176]	max(0.525957,0.388084)	
[-4.594886,8.508279]			
[-4.751445,8.603235]	[-5.197089,8.872567]	max(0.00785,0.021391)	
[-4.865385,8.673918]			
[-4.950393,8.725095]	[-5.197506,8.873307]	max(0.000417,0.00074)	[-5.197506,8.873307]
[-5.197506,8.873307]			
[-5.197506,8.873307]	[-5.197089,8.872567]	max(0.000417,0.00074)	[-5.197089,8.872567]

Méthodologie par l'exemple

```
while (1) {
xn1 = -0.4375 * x1 + 0.0625 * x2 + 0.2652 * x3 + 0.1 * u1;
xn2 = 0.0625 * x1 + 0.4375 * x2 + 0.2652 * x3 + 0.1 * u2;
xn3 = -0.2652 * x1 + 0.2652 * x2 + 0.375 * x3 + 0.1 * u3;
x1 = xn1;x2 = xn2;x3 = xn3;
} [-5.197505,8.873306]
```

Itération: 15

x	$\Upsilon_A(\overrightarrow{y})$	$\|y_i - y_{i-1}\|$	**Seuils** ($\delta = 10^{-3}$)
[1.000000,2.000000]	[1.000000,2.000000]		
[-0.447300,5.716500]			
[-2.291255,6.573381]	[6.280857,6.830145]	max(5.280857,4.830145)	
[-3.038029,7.492492]			
[-3.695558,7.871720]	[-4.663282,8.463092]	max(10.944139,2.367053)	
[-4.084503,8.185954]			
[-4.386442,8.369221]	[-5.189239,8.851176]	max(0.525957,0.388084)	
[-4.594886,8.508279]			
[-4.751445,8.603235]	[-5.197089,8.872567]	max(0.00785,0.021391)	
[-4.865385,8.673918]			
[-4.950393,8.725095]	[-5.197506,8.873307]	max(0.000417,0.00074)	[-5.197506,8.873307]
[-5.197506,8.873307]			
[-5.197506,8.873307]	[-5.197089,8.872567]	max(0.000417,0.00074)	[-5.197089,8.872567]
[-5.197567,8.873341]			
[-5.197616,8.873348]	[-5.198638,8.849716]	max(0.001549,0.022851)	

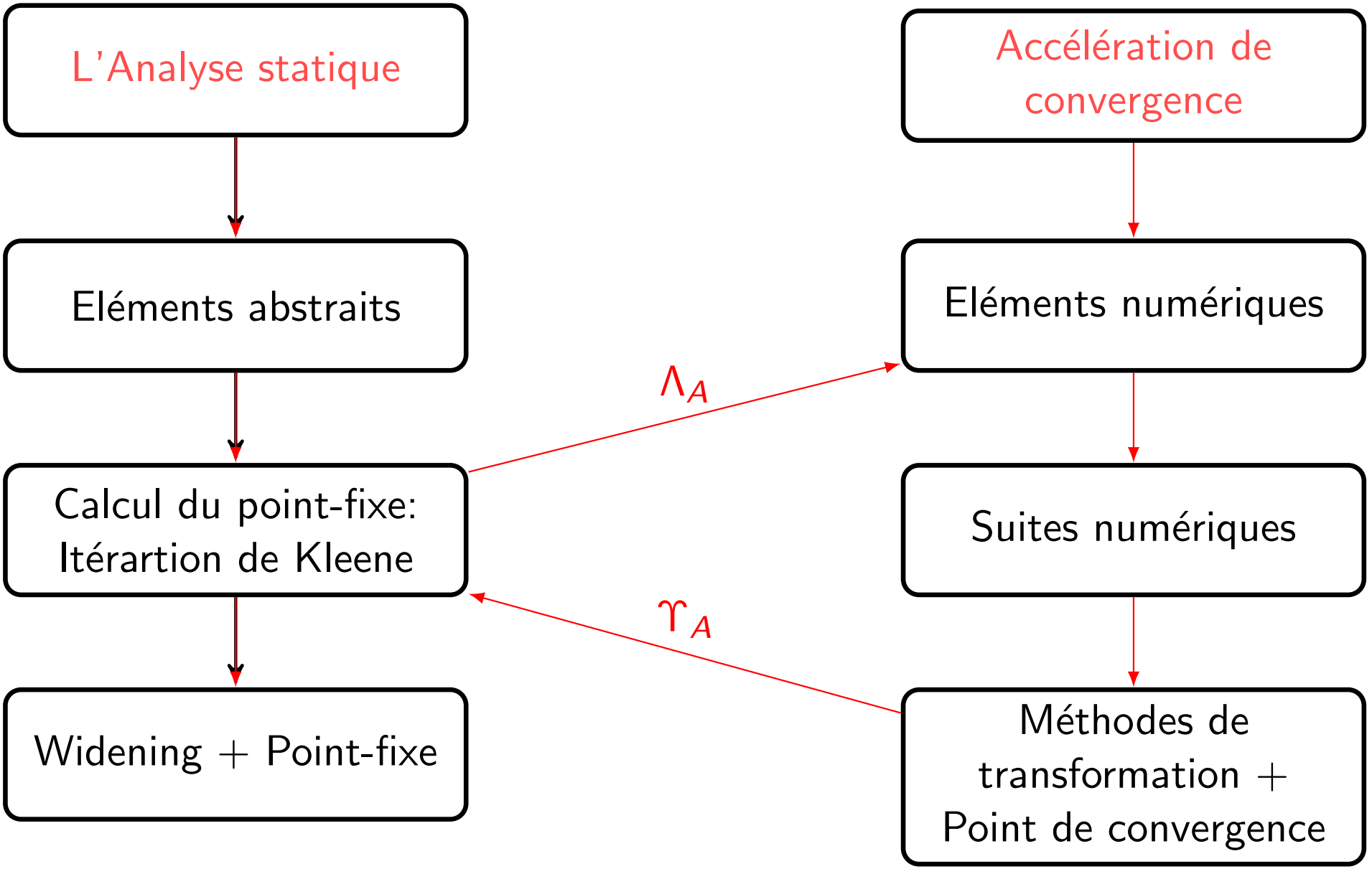
Méthodologie par l'exemple

```
while (1) {
xn1 = -0.4375 * x1 + 0.0625 * x2 + 0.2652 * x3 + 0.1 * u1;
xn2 = 0.0625 * x1 + 0.4375 * x2 + 0.2652 * x3 + 0.1 * u2;
xn3 = -0.2652 * x1 + 0.2652 * x2 + 0.375 * x3 + 0.1 * u3;
x1 = xn1;x2 = xn2;x3 = xn3;
} [-5.197505,8.873306] ↷ [-5.197598,8.873362]
```

Itération: 16

x	$\Upsilon_A(\overrightarrow{y})$	$\|y_i - y_{i-1}\|$	**Seuils** ($\delta = 10^{-3}$)
[1.000000,2.000000]	[1.000000,2.000000]		
[-0.447300,5.716500]			
[-2.291255,6.573381]	[6.280857,6.830145]	max(5.280857,4.830145)	
[-3.038029,7.492492]			
[-3.695558,7.871720]	[-4.663282,8.463092]	max(10.944139,2.367053)	
[-4.084503,8.185954]			
[-4.386442,8.369221]	[-5.189239,8.851176]	max(0.525957,0.388084)	
[-4.594886,8.508279]			
[-4.751445,8.603235]	[-5.197089,8.872567]	max(0.00785,0.021391)	
[-4.865385,8.673918]			
[-4.950393,8.725095]	[-5.197506,8.873307]	max(0.000417,0.00074)	[-5.197506,8.873307]
[-5.197506,8.873307]			
[-5.197506,8.873307]	[-5.197089,8.872567]	max(0.000417,0.00074)	[-5.197089,8.872567]
[-5.197567,8.873341]			
[-5.197616,8.873348]	[-5.198638,8.849716]	max(0.001549,0.022851)	
[-5.197598,8.873362]			

Récapitulatif



Les fonctions *Extraction* et *combination*

Propriété 01: Transformation localement sûre

Soient $\langle A, \sqsubseteq_A \rangle$ un ensemble ordonné et $\langle D, d \rangle$ un espace métrique.
Soient $\phi : A \to D$ et $\psi : D \to A$ deux fonctions. Les fonctions ϕ et ψ forment une transformation localement sûre ssi: $\forall X \in A, X \sqsubseteq_A \psi \circ \phi(X)$.

Propriété 02: Transformation asymptotiquement sûre

Soient $\langle A, \sqsubseteq_A \rangle$ un ensemble ordonné et $\langle D, d \rangle$ un espace métrique.
Les fonctions $\phi : A \to D$ et $\psi : D \to A$ forment une transformation asymptotiquement sûre ssi:
Quelque soit la suite monotone et convergente (S_n), tel que
$\forall n \in \mathbb{N}, S_n \in A$, avec $\lim_{n \to +\infty} S_n = \bigcup S_n = S$.
On a que: si $\lim_{n \to +\infty} \phi(S_n) = X$ alors $S \sqsubseteq_D \psi(X)$.

Les fonctions *Extraction* et *combination*

Définition:

Soient $\langle A, \sqsubseteq_A \rangle$ un domain abstrait, et $\langle D, d \rangle$ un espace métrique. Les fonctions $\Lambda_A : A \to D$ et $\Upsilon_A : D \to A$ sont les fonctions *Extraction* et *combination*, respectivement, *ssi*: ils forment une transformation localement et asymptotiquement sûre.

Exemple: Les fonction Λ et Υ pour le domaine des intervalles.

$$\Lambda_{D_I}: \begin{cases} \mathrm{D}_I & \to \mathbb{R}^2 \\ [\mathrm{x},\mathrm{y}] & \to (x, y) \end{cases} \qquad \Upsilon_{D_I}: \begin{cases} \mathrm{R}^2 & \to D_I \\ (\mathrm{x},\mathrm{y}) & \to [x, y] \end{cases}$$

Pour changer de domaine il suffit de changer ces deux fonctions.

Représentation sous forme de DBM: exemple

The octagon constraints.

$$\begin{cases} x - y \leq 3 \\ x + y \leq 0 \\ -x - y \leq -3 \\ x \leq 1 \end{cases}$$

The potential constraints.

$$\begin{cases} x_2 - y_2 \leq 3 & \textit{and} & y_1 - x_1 \leq 3 \\ x_2 - y_1 \leq 0 & \textit{and} & y_2 - x_1 \leq 0 \\ x_1 - y_2 \leq -3 & \textit{and} & y_1 - x_2 \leq -3 \\ x_2 - x_1 \leq 2 \end{cases}$$

The DBM.

$$\begin{array}{c} \\ x_1 \\ x_2 \\ y_1 \\ y_2 \end{array} \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & y_1 & y_2 \end{array} \\ \begin{pmatrix} +\infty & 2 & 3 & 0 \\ +\infty & +\infty & -3 & +\infty \\ +\infty & 0 & +\infty & +\infty \\ -3 & 3 & +\infty & +\infty \end{pmatrix} \end{array}$$

Les fonctions *Extraction* et *combination*: Octogone

Notons que $w = 2v$ avec v le nombre de variables du programme.

Définition de Λ pour le domaine des octogones

$$\Lambda_{oct} : \begin{cases} oct & \to & R^{w^2} \\ \begin{pmatrix} i_{11} & \dots & i_{1w} \\ \vdots & \ddots & \vdots \\ i_{w1} & \dots & i_{ww} \end{pmatrix} & \mapsto & \left(i_{11}, \dots, i_{1w}, \dots, i_{w1}, \dots, i_{ww}\right) \end{cases}$$

Définition de Υ pour le domaine des octogones

$$\Upsilon_{oct} : \begin{cases} R^{w^2} & \to & oct \\ \left(x_1, \dots, x_w, \dots, x_{w^2}\right) & \mapsto & \begin{pmatrix} x_1 & \dots & x_w \\ \vdots & \ddots & \vdots \\ x_{w^2-w} & \dots & x_{w^2} \end{pmatrix} \end{cases}$$

Expérimentation: Contraction

```
while (1) {
xn1 = -0.4375 * x1 + 0.0625 * x2 + 0.2652 * x3 + 0.1 * u1;
xn2 = 0.0625 * x1 + 0.4375 * x2 + 0.2652 * x3 + 0.1 * u2;
xn3 = -0.2652 * x1 + 0.2652 * x2 + 0.375 * x3 + 0.1 * u3;
x1 = xn1;
x2 = xn2;
x3 = xn3;
}
```

	Contraction
Itération de Kleene	127
L'ε-Algorithme	16
L'ε-Algorithme Vectoriel	17

Expérimentation: Butterworth1

```
void main () {
 x1 = 0; y = 0; xn1 = 1;
while (1) {
  xn1 = 0.9048 *x1 + 0.9524 *u;
  y = 0.09524  *x1  + 0.04762*u;
  x1 = xn1;
}  }
```

	Butterworth1
Itération de Kleene	346
L'ε-Algorithme	08
L'ε-Algorithme Vectoriel	10

Expérimentation: Butterworth2

```
void main () {
x1 = 0;x2 = 0;u=1;
while (1) {
xn1 = 0.8027 *x1 + -0.07314 *x2 + 0.9314 *u;
xn2 = 0.07314 *x1 + 0.8053 *x2 + 0.04657*u;
y = 0.004657 *x1 + 0.09977 *x2 + 0.002328*u;
x1 = xn1;
x2 = xn2;
} }
```

	Butterworth2
Itération de Kleene	83
L'ε-Algorithme	47
L'ε-Algorithme Vectoriel	26

Expérimentation

	Nombre d'itérations			Distance
Programme	Kleene	Accel.	Ratio	Distance
`contraction.c`	127	21	1/6	8.10^{-14}
`butter1.c`	346	10	1/24	3.10^{-14}
`butter2.c`	181	21	1/9	2.10^{-14}

Table: Les résultats obtenus en utilisant le domaine abstrait des octogones (avec $\delta = 10^{-3}$).

Expérimentation

```
void main () {
x1 = 0;x2 = 0;u=1;
while (1) {
xn1 = 0.8027 *x1 + -0.07314 *x2 + 0.9314 *u;
xn2 = 0.07314 *x1 + 0.8053 *x2 + 0.04657*u;
y = 0.004657 *x1 + 0.09977 *x2 + 0.002328*u;
x1 = xn1;
x2 = xn2;
} }
```

Variables	Itération de Kleene	l'ε-Algorithme Vectoriel		
		$\delta = 10^{-3}$	$\delta = 10^{-4}$	$\delta = 10^{-5}$
x_1	70	7	9	22
x_2	83	26	23	17
y	83	26	23	19

Conclusion et perspective

Conclusion

Améliorer l'itération de Kleene:

- Gagner de la précision du calcul.
- Utilisation des méthodes d'accélération de convergence des suites numériques.
- Combiner avec la méthode du widening: le cas d'une suite non convergente.

Perspectives

- Étendre notre technique d'accélération à d'autres domaines abstraits.
- Inclure notre technique d'accélération à Interproc.
- Expérimenter d'autres algorithmes d'accélération de convergence.
- Comparer avec d'autres méthodes de calcul du point-fixe.