Résolution de contraintes sur les nombres à virgule flottante par une approximation sur les réels*

Mohammed Said Belaid, Claude Michel, Michel Rueher

I3S-CNRS, Pôle MDSC, équipe CeP

*Ce travail a été partiellement financé par l'ANR, programme SESUR, projet CAVERN (ANR-07-SESU-003).

Introduction

- Dégâts causés par l'utilisation sans précaution de l'arithmétique des flottants:
  - Explosion de la fusée Ariane V.
    - Conséquence : 7 milliards de Dollars.
    - Cause : Conversion d'un nombre flottant sur 64 bits vers un entier sur 16 bits.
  - L'échec de l'interception d'un missile par Patriot.
    - Conséquence : 25 morts et 100 blessés.
    - Cause : une erreur de l'ordre de 10^{-7} dans le calculs de temps.

Introduction

- Vérification de programmes avec du calcul flottant.
  - ASTRÉE : [Miné et al.2004] Garantir l'absence des erreurs d'exécution.
  - Fluctuat : [Goubault et al.2002] Estimation de l'erreur d'arrondi.

Introduction

- Les outils de vérification basés sur la PPC sont peu utilisés pour les programmes avec du calcul flottant.
  - Euclide : Génération de cas de test
  - CPBPV : Validation des programmes vis-à-vis leurs spécifications.

Introduction

$\mathbb{F}$

$x \oplus 16 = 16$

$\Downarrow$

$0 \leq x \leq 9{,}5 * 10^{-7}$

$\mathbb{R}$

$x + 16 = 16$

$\Downarrow$

$x = 0$

Introduction

- Les méthodes existantes adaptent les algorithmes conçues principalement pour les réels aux spécificité des flottants. [Michel et al. 2001, Botella et al.2005]
  - Box-Concistency dédié aux flottants.
  - 2B-Concistency dédié aux flottants.

La méthode proposée

Transformer le problème de résolution de contraintes sur les flottants vers la résolution de contraintes sur les réels.

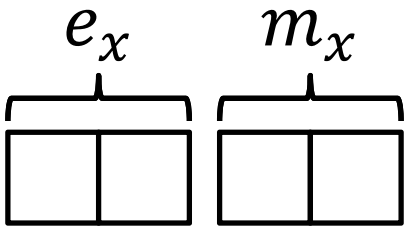
Plan

- La méthode proposée
  - Exemple illustratif
  - Les étapes de la méthode
- La transformation de contraintes
- Exemple d’application
- Perspectives

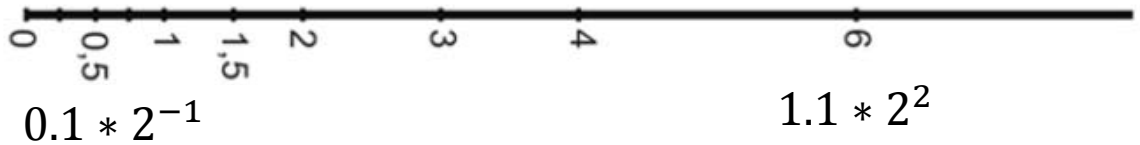
La méthode proposée

(Exemple illustratif)

- Type réduit de flottants:



$$x = m_x 2^{e_x}$$



$0.1 * 2^{-1}$ $1.1 * 2^{2}$

  - Le mode d'arrondi est fixé vers $-\infty$.
  - Ulp est le pas entre deux flottants.

$$ulp(x) = 2^{e_x - p + 1}$$

La méthode proposée

(Exemple illustratif)

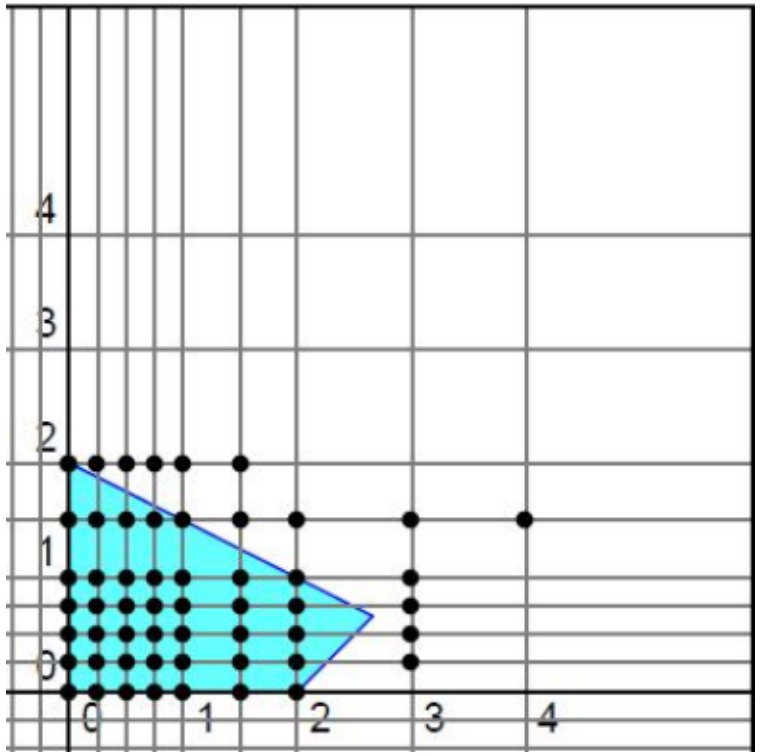
- Soit le système de contraintes suivant
  - Avec *x* et *y* des variables de type flottant introduit

$$\begin{cases} x & \geq 0 \\ y & \geq 0 \\ (x \oplus y) \oplus y & \leq 4 \\ x \ominus y & \leq 2 \end{cases}$$

La méthode proposée

(Exemple illustratif)

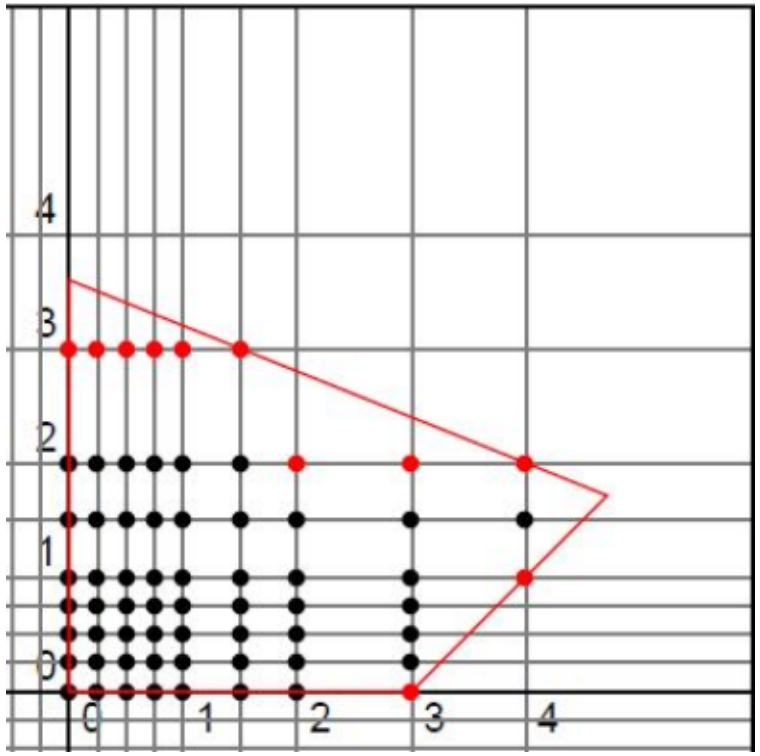
- Les points **noirs** sont solutions du système sur les flottants. L'espace en **bleu** représente les solution du système sur les réels.
- L'espace des réels ne contient pas les solutions sur les nombres flottants.
- On cherche donc une sur-approximation qui contient toutes les solutions.



La méthode proposée

(Exemple illustratif)

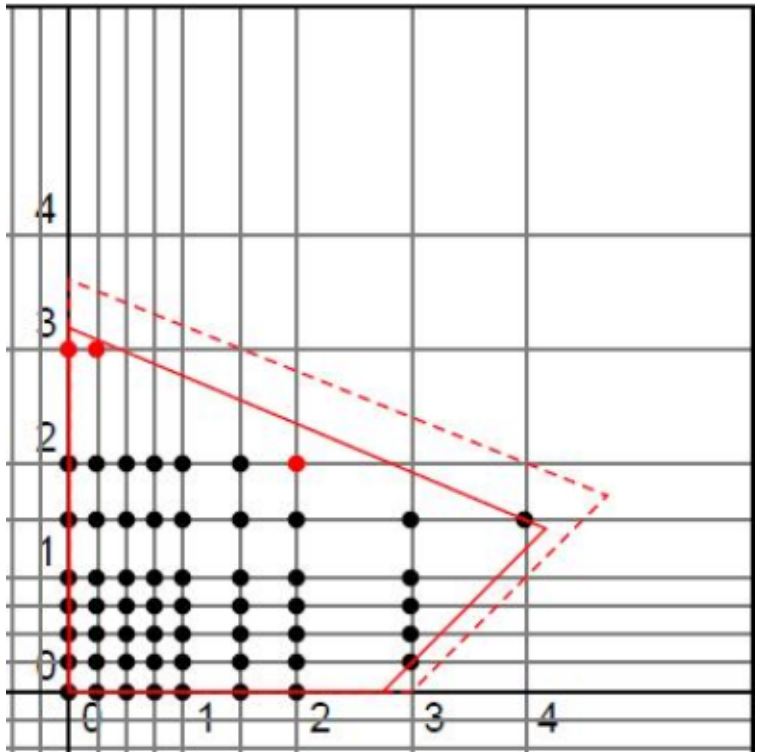
- Une première méthode d'approximation:
- Les points **noirs** sont des solutions du système sur les flottants. Les points en **rouge** sont les points non-solutions.
- L'espace d'approximation contient aussi des points non-solutions.
- Nous cherchons à réduire l'espace non-solution.



La méthode proposée

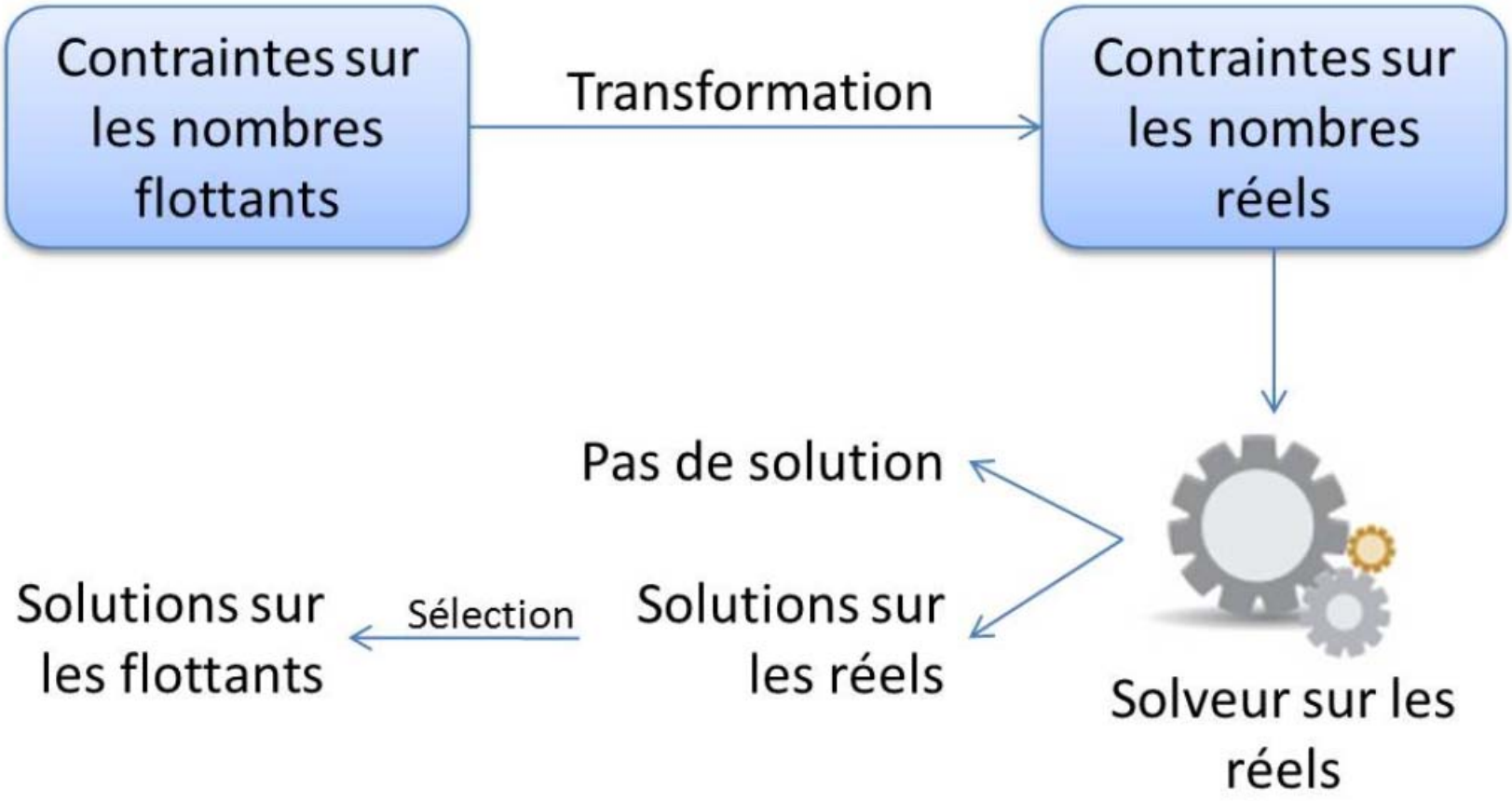
(Exemple illustratif)

- Une deuxième méthode d'approximation plus fine:
- Les points **noirs** sont des solutions du système sur les flottants. Les points en **rouge** sont les points non-solutions.
- L'espace des points non-solutions est réduit avec cette méthode.



La méthode proposée

(Les étapes)



La méthode proposée

(Les étapes)

1. Transformation de contraintes sur les flottants vers des contraintes sur les réels.
    - L'approximation doit être conservative de solutions sur les flottants.
    - L'approximation peut contenir des éléments non-solutions.
    - Nous cherchons à réduire l'espace de non-solution au strict minimum.

La méthode proposée

(Les étapes)

2. Les contraintes obtenues sont ensuite résolue par un solveurs sur les réels.
3. Une autre étape d'énumération est requise pour choisir les solutions du système de contrainte sur les flottants.

Transformation de contraintes sur les flottants

- Approximation des fonctions sur les flottants par des intervalles sur les réels

$$f(x) \geq g(x)$$

$$[f_{inf}(x), f_{sup}(x)] \geq [g_{inf}(x), g_{sup}(x)]$$

f, g sont des fonctions sur les flottants

$f_{inf}, f_{sup}, g_{inf}, g_{sup}$ sont des fonctions sur les réels

Transformation de contraintes sur les flottants

- Pour chaque opération de base sur les flottants nous devons définir un intervalle sur les réels.
- Nous utilisons ensuite l'arithmétique des intervalles pour obtenir l'approximation des expressions arithmétiques composées.

Transformation de contraintes sur les flottants

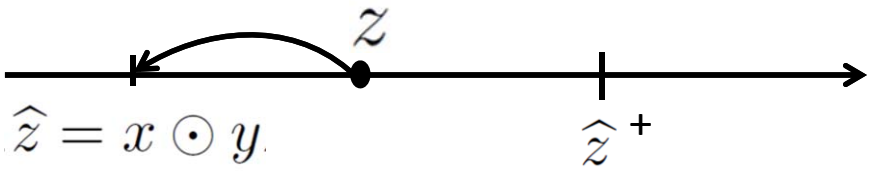
Résultat flottant $\widehat{z} = x \odot y$ Résultat réelle $z = x \cdot y$

$\cdot \in \{+, -, \times, /\}$

Mode d'arrondi \ z	-Max ≤ z < -Min Négatif normalisé	-Min ≤ z < 0 Négatif dénormalisé	0 ≤ z < Min Positif dénormalisé	Min ≤ z ≤ Max Positif normalisé
Vers -∞	$[(1+2^{-p+1}) * z, z]$	$[z - \min, z]$	$[z - \min, z]$	$[\frac{1}{1+2^{-p+1}} * z, z]$
Vers 0	$[z, \frac{1}{1+2^{-p+1}} * z]$	$[z, z + \min]$	$[z - \min, z]$	$[\frac{1}{1+2^{-p+1}} * z, z]$
Vers +∞	$[z, \frac{1}{1+2^{-p+1}} * z]$	$[z, z + \min]$	$[z, z + \min]$	$[(1+2^{-p+1}) * z, z]$
Au plus proche	$[\frac{1+2^{-p+1}}{1+2^{-p}} * z, \frac{1-2^{-p}}{1-2^{-p+1}} * z]$	$[z - \frac{min}{2}, z + \frac{min}{2}]$	$[z - \frac{min}{2}, z + \frac{min}{2}]$	$[\frac{1-2^{-p}}{1-2^{-p+1}} * z, \frac{1+2^{-p+1}}{1+2^{-p}} * z]$

Transformation de contraintes sur les flottants

- Pour un mode d'arrondi vers -∞
- Et un résultat positif normalisé



$$\widehat{z} \leq z < \widehat{z}^+$$

$$\widehat{z} \leq z < \widehat{z} + ulp(\widehat{z})$$

$$z - ulp(\widehat{z}) < \widehat{z} \leq z$$

Transformation de contraintes sur les flottants

- On calcule une approximation pour l'erreur relative.

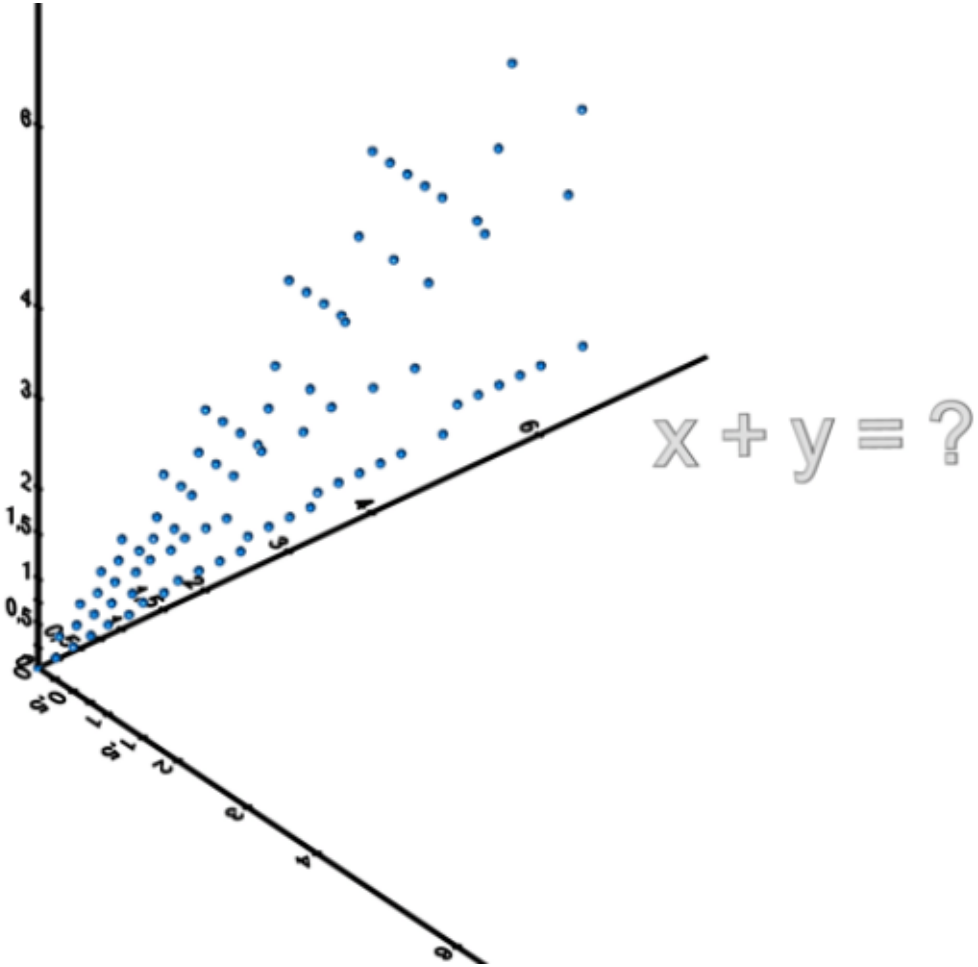
$$0 \leq \epsilon < \frac{ulp(\widehat{z})}{\widehat{z} + ulp(\widehat{z})}$$

$$z = m_z 2^{e_z}$$

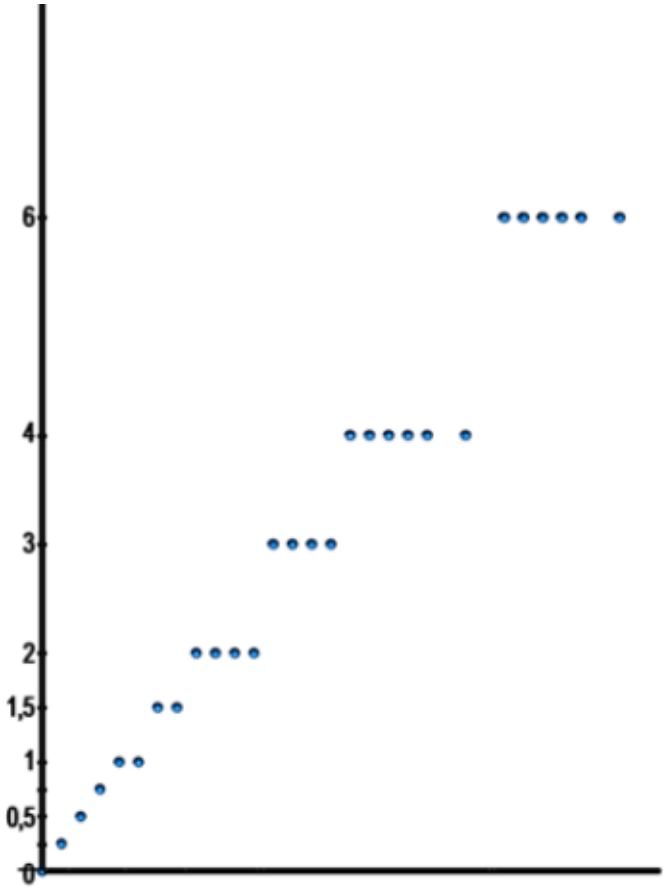
$$0 \leq \epsilon < \frac{ulp(\widehat{z})}{\widehat{z} + ulp(\widehat{z})} = \frac{2^{-p+1}}{m_z + 2^{-p+1}} < \frac{2^{-p+1}}{1 + 2^{-p+1}}$$

$$\frac{1}{1 + 2^{-p+1}} z < \widehat{z} \leq z$$

Les différents niveaux d’approximation

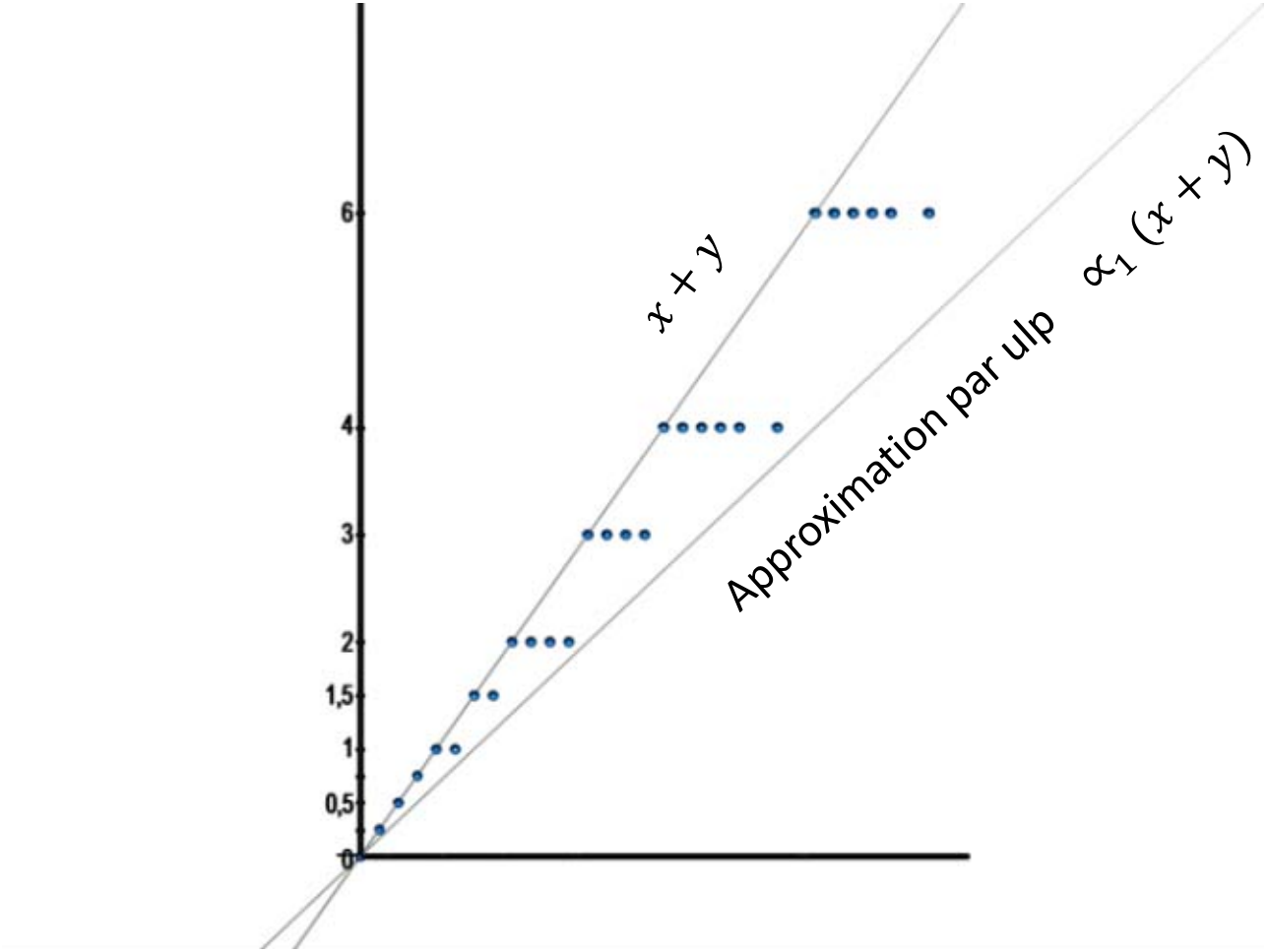


Les différents niveaux d’approximation

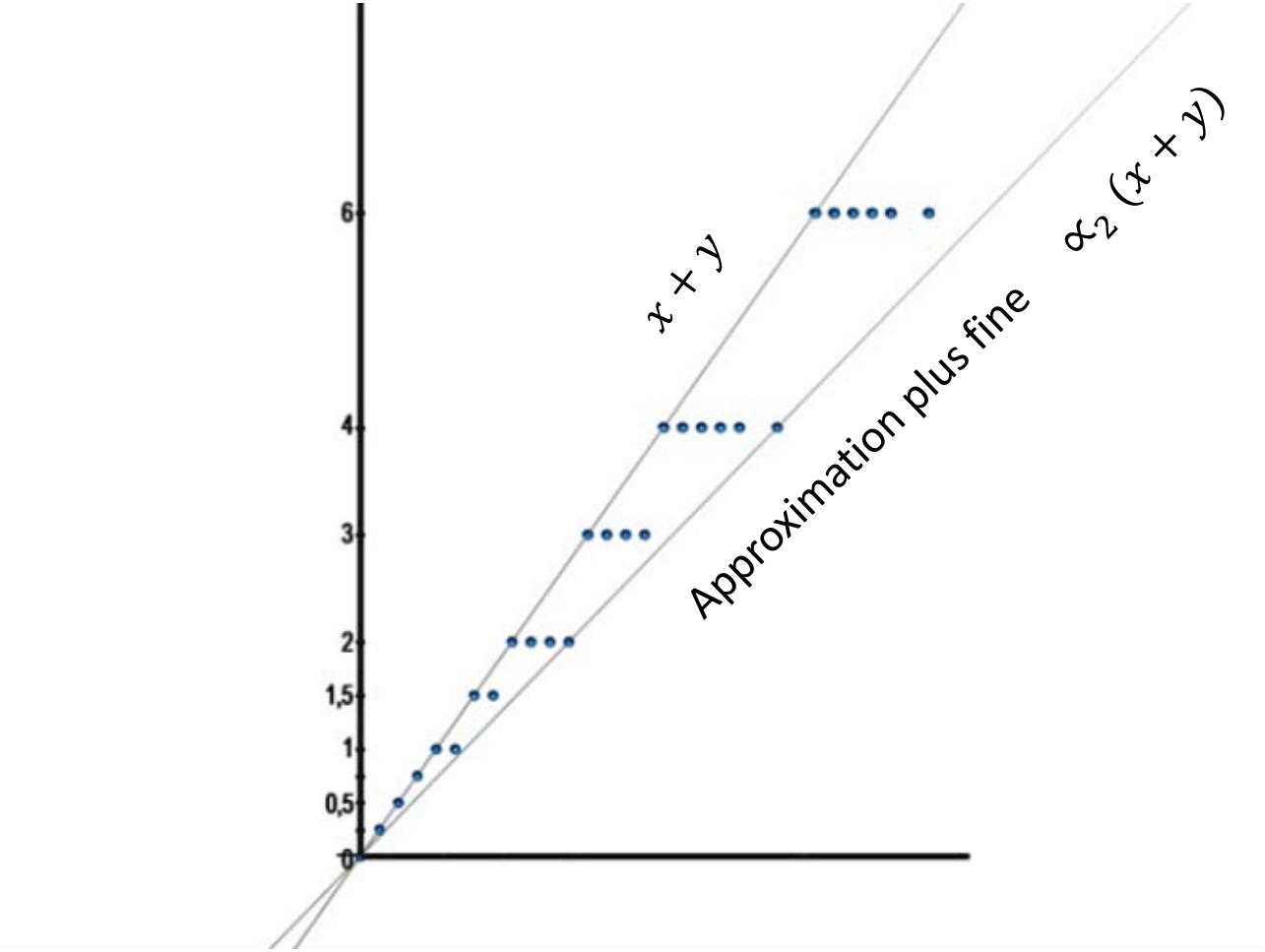


Vue profil des
résultats de (x + y)

Les différents niveaux d'approximation



Les différents niveaux d'approximation



Transformation de contraintes sur les flottants

2ème niveau d'approximation pour l'addition

- La méthode consiste à chercher le pire cas qui peut réellement figurer.
- Pour l'addition :
  - Si le résultat est positif et le mode d'arrondi est vers -∞
  - $(x \oplus y = x)$ est le pire cas d'arrondi correspondant.

$$\frac{1}{1+(2^{-p+1}-2^{-2p+1})} \times (x+y) \leq x \oplus y \leq x+y$$

Exemple d'application

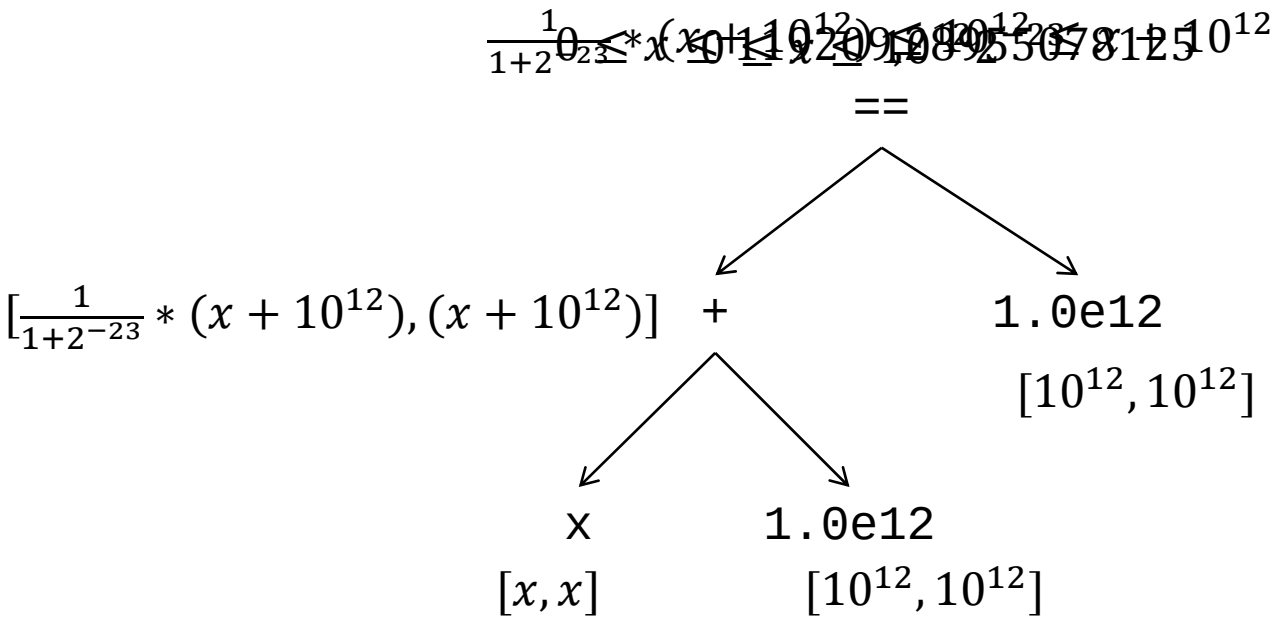
```
float foo(float x){ // [Botella et al.2004]
      float y = 1.0e12, z = 0.0;
1.    if (x > 1000)
2.          z = x + y;
3.    if (z == y)
4.          // ..
      ;
}
```

1 -> 2 -> 3 -> 4 :
(x > 1000) (x + 1.0e12 == 1.0e12)

un solveur sur les réels ne donne aucune solutions. Par contre plusieurs solutions sont possibles sur les flottants.

Exemple d'application

Approximation de la contrainte **`(x + 1.0e12 == 1.0e12)`**



Exemple d'application

(x + 1.0e12 == 1.0e12) $\longrightarrow$ $0 \leq x \leq 119209{,}28955078125$

(x > 1000) $\longrightarrow$ $x > 1000$

$1000 < x \leq 119209{,}28955078125$

Perspectives

- Implémentation de la phase de transformation de contraintes sur les flottants. ✅
- Choix du solveurs pour la résolution des contraintes. ✅
- Application sur les méthodes de vérification de programmes basées sur la PPC. ❌
  - Génération de cas de test pour des comportements inattendus des programmes.
  - Une connexion à CPBPV est envisagée.
- Approximation plus fine pour la multiplication par des constantes. ✅
- Approximation des fonctions transcendantal. ❌

Merci pour votre attention

Question?