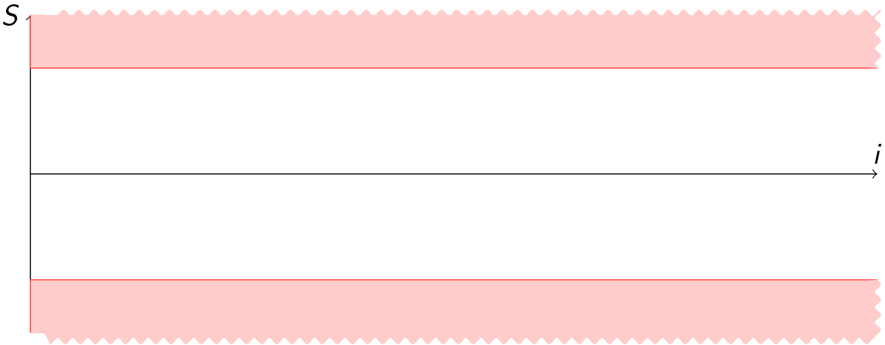
Static Analysis of Numerical Programs Constrained Affine Sets-Abstract Domain

Khalil Ghorbal
CMU

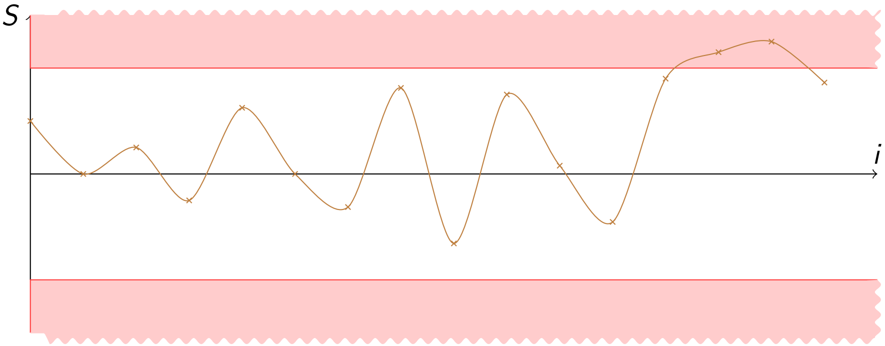
Joint work with Éric Goubault and Sylvie Putôt

NASA AMES
March 12th, 2013

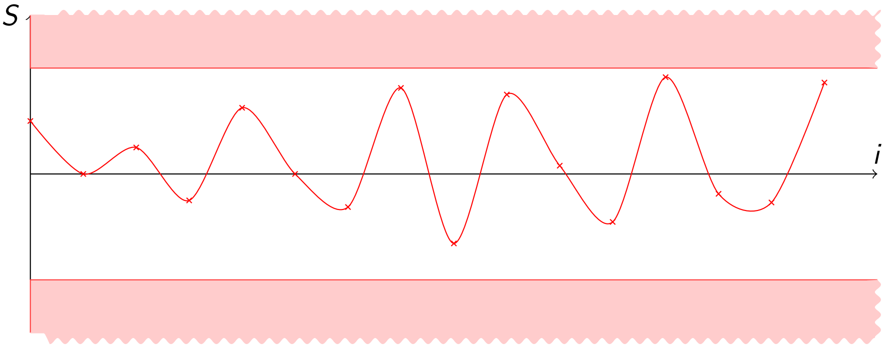
Abstract Interpretation : Intuitions



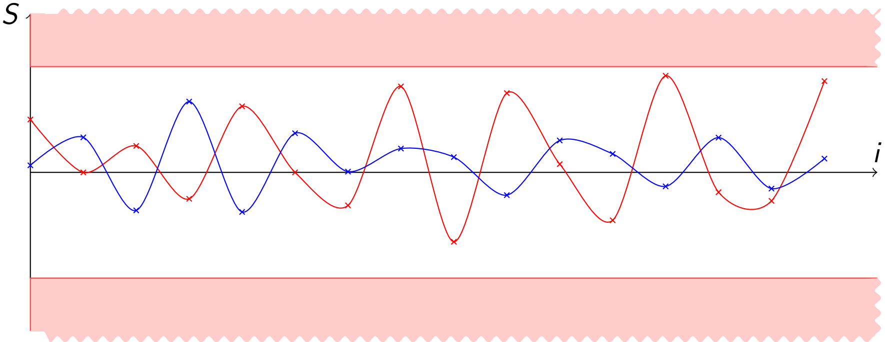
Abstract Interpretation : Intuitions



Abstract Interpretation : Intuitions

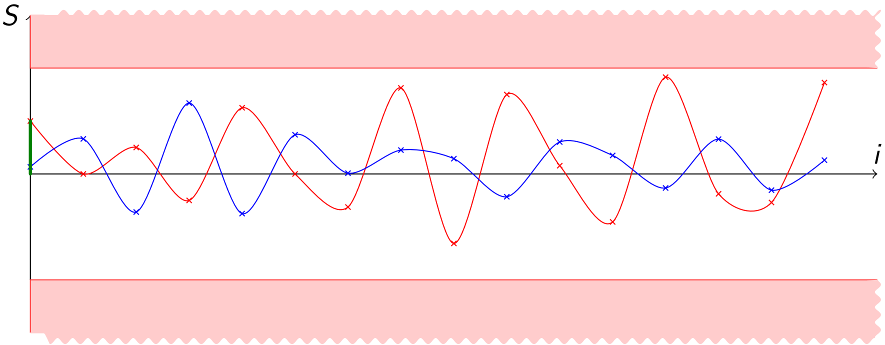


Abstract Interpretation : Intuitions

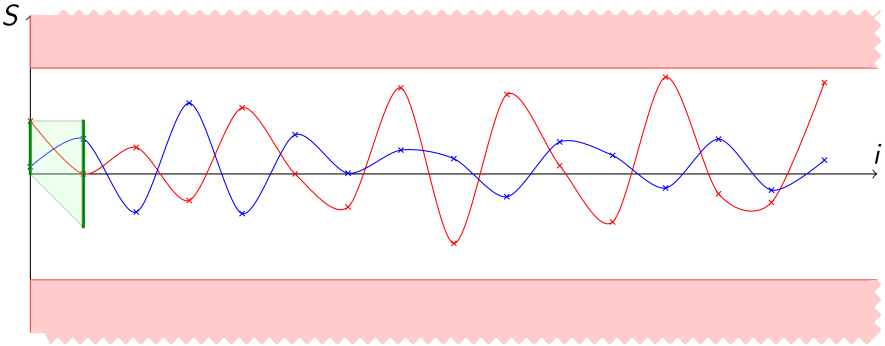


- What about the missed bugs ? are they severe ?

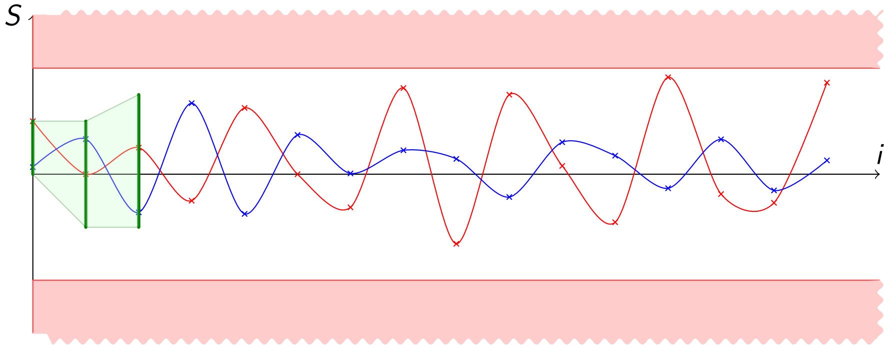
Abstract Interpretation : Intuitions



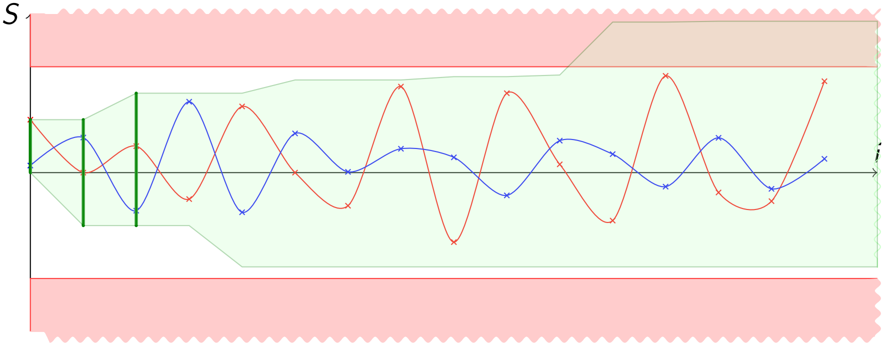
Abstract Interpretation : Intuitions



Abstract Interpretation : Intuitions

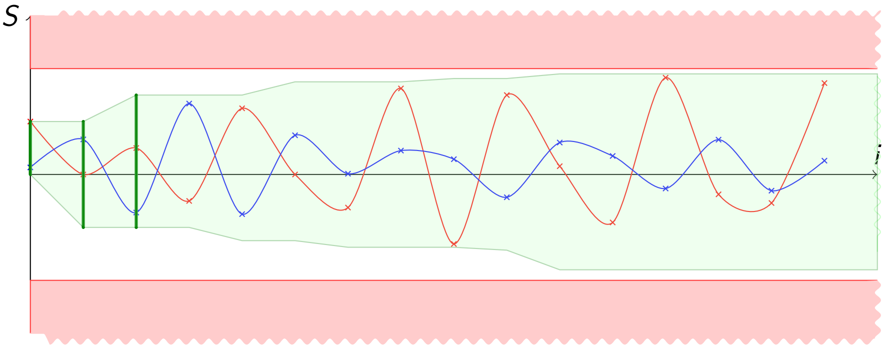


Abstract Interpretation : Intuitions



- Over-approximation may lead to **false alarms**.

Abstract Interpretation : Intuitions



➻ Accurate over-approximation gives a safety **proof**.

Famous bugs

Examples

- 1982, The Vancouver stock exchange: after 22 months the index had fallen to $524,811$ instead of $1098,811$
- 1985, Therac 25 (radiation therapy machine) : 5 patients killed (overdoses of radiation)
- 1991, The Patriot Missile: 28 soldiers killed
- 1996, Ariane 5: more than 1 billion $ gone in 40 seconds

E. Dijkstra (1972)

Program testing can be used to show the presence of bugs, but never to show their absence!

ESA Project - Automated Transfer Vehicle (ATV)

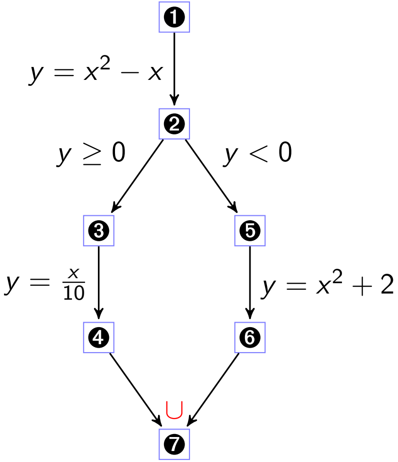
Jules Verne (Key dates)
9th Mars 2008 launching
3th April 2008 docking to ISS
11th September 2008, undocking
29th September 2008, end of mission

✓ Publication in *DAta Systems In Aerospace* (DASIA) 2009

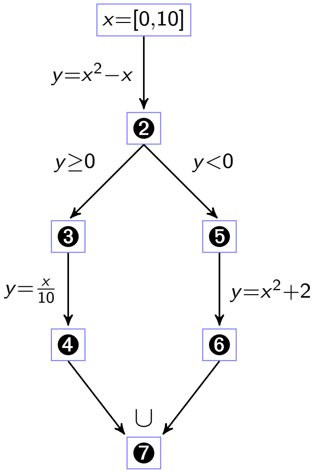
All existing abstract domains fail to handle precisely **normalized quaternions**

Detailed example

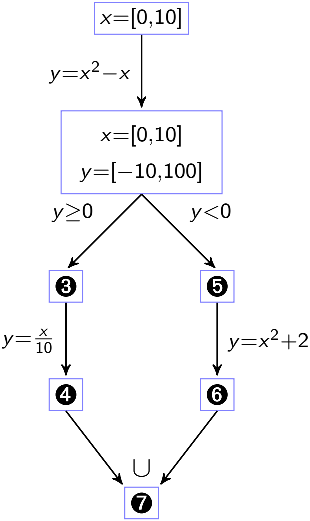
```
begin
x = [0,10]; ❶
y = x*x - x ❷
if (y >= 0)  ❸ then
y = x / 10; ❹
else  ❺
  y = x*x + 2; ❻
done; ❼
end
```



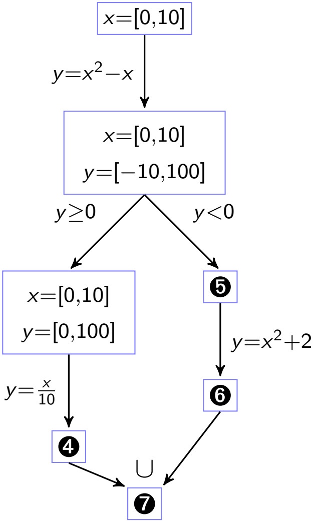
Forward Propagation



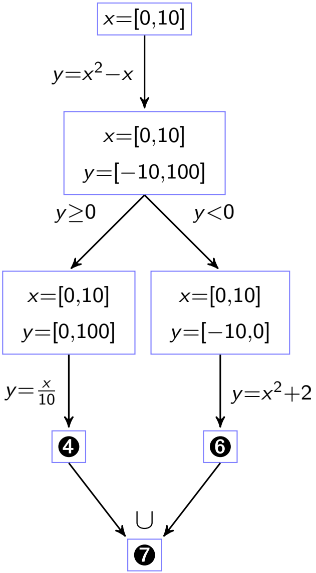
Forward Propagation



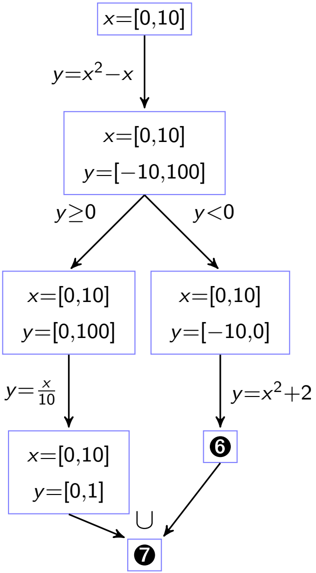
Forward Propagation



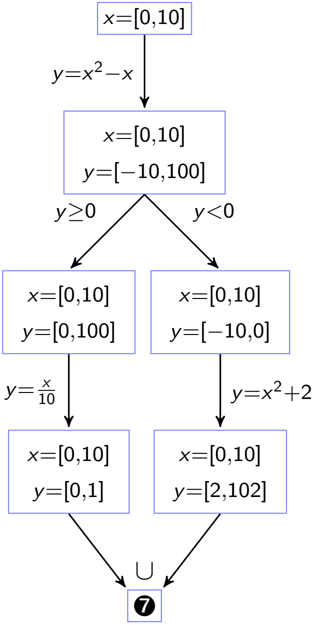
Forward Propagation



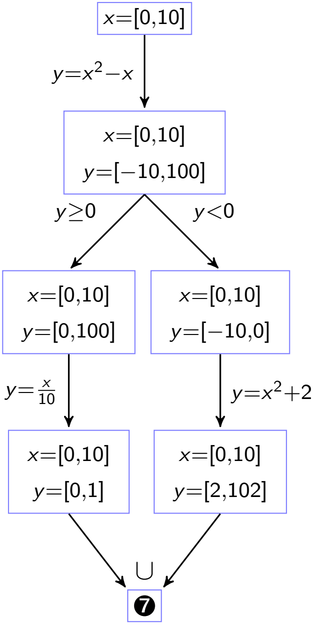
Forward Propagation

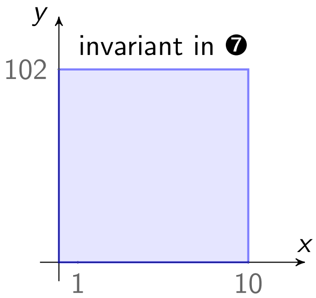


Forward Propagation

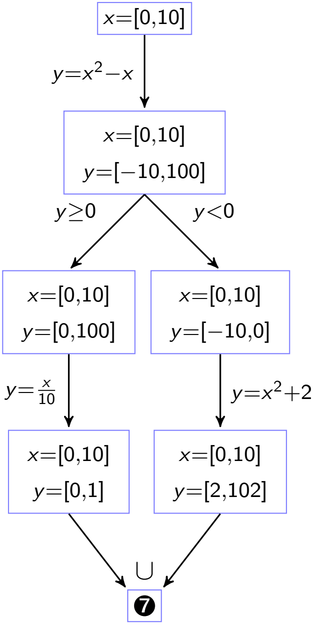


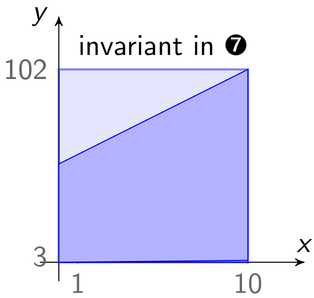
Forward Propagation



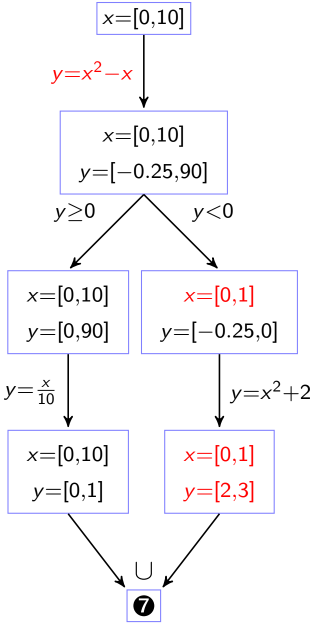


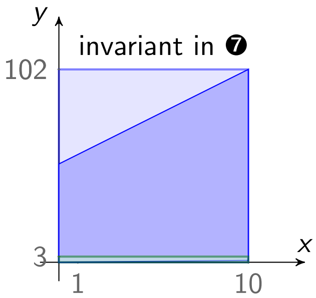
Forward Propagation



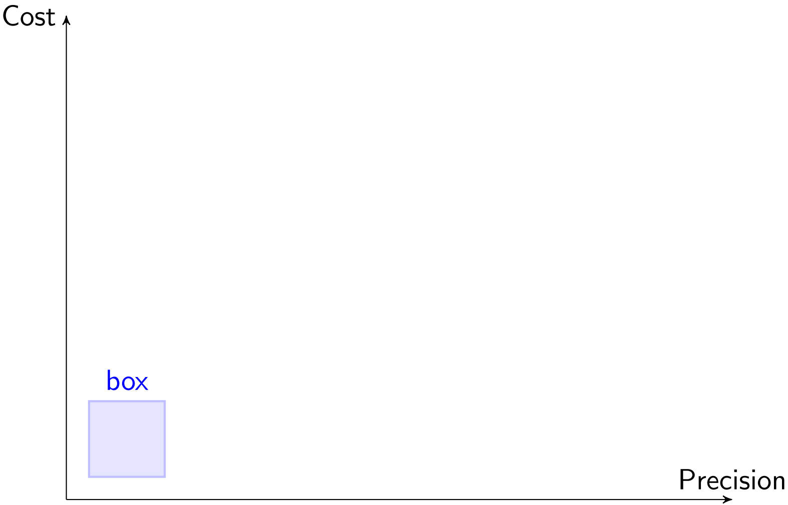


Forward Propagation

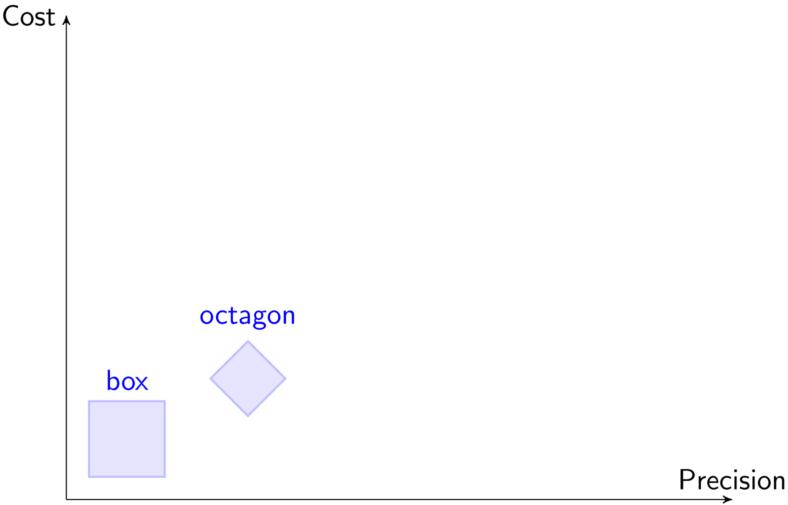




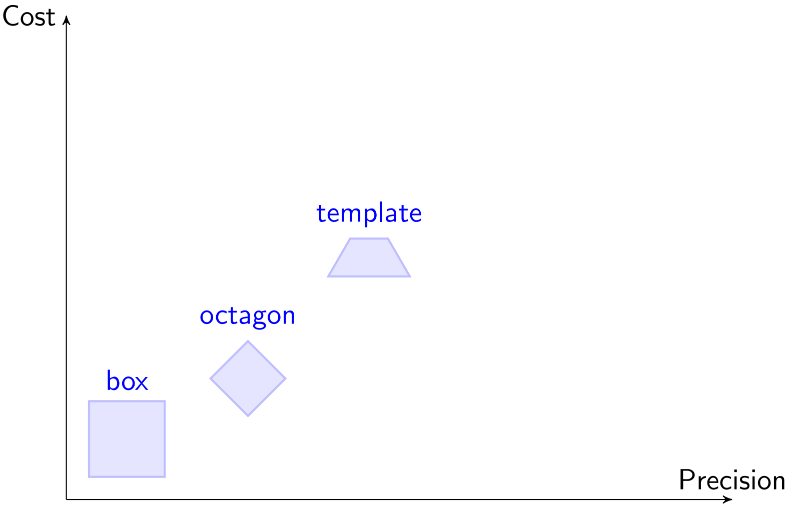
Precision Cost Trade-off



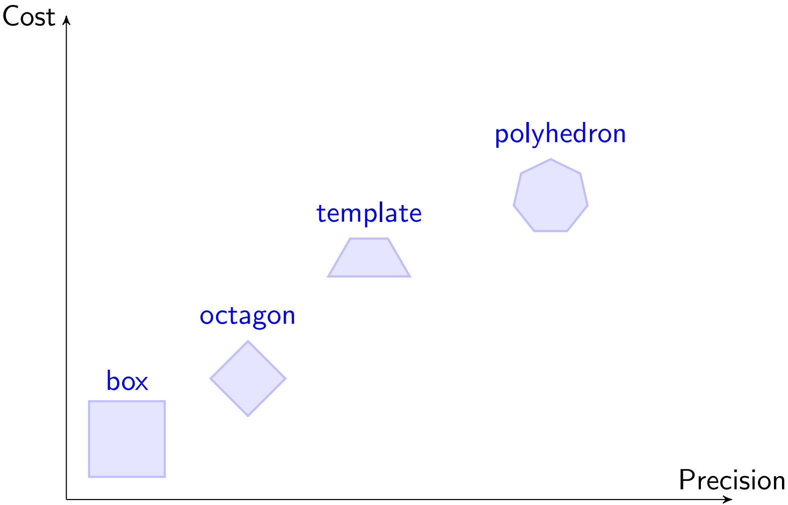
Precision Cost Trade-off



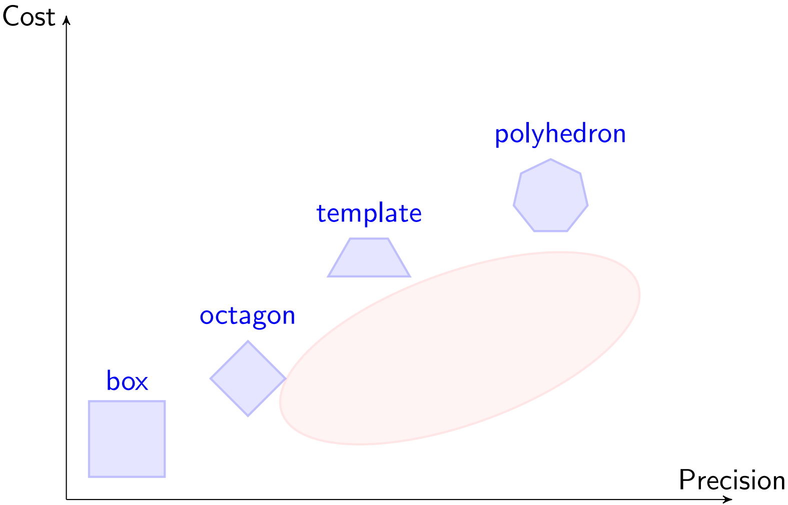
Precision Cost Trade-off



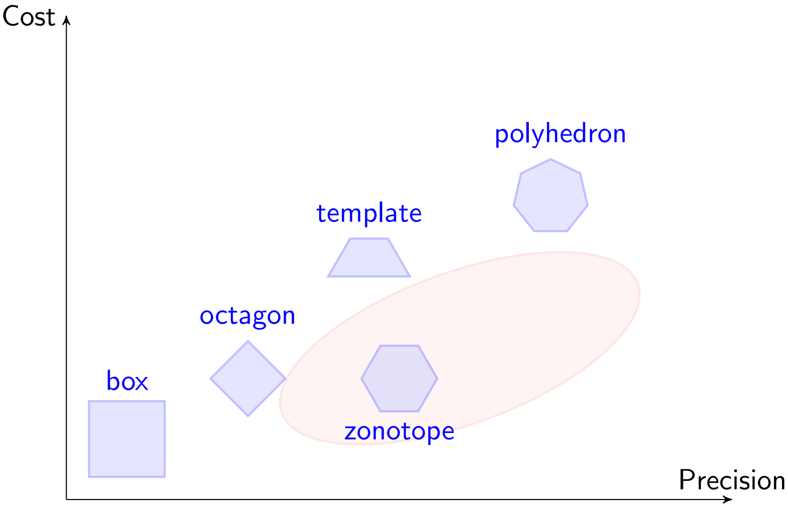
Precision Cost Trade-off



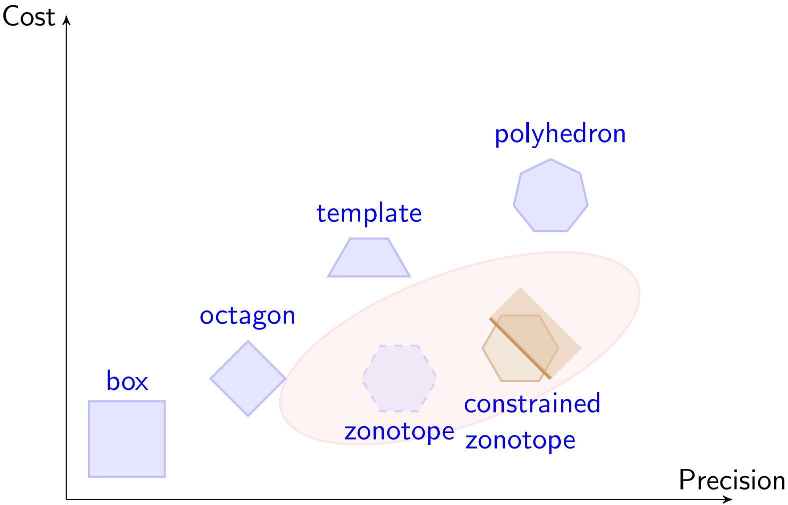
Precision Cost Trade-off



Precision Cost Trade-off



Precision Cost Trade-off



Outlines

Formal Verification Approaches

Formal Verification Approaches

- Hoare 1969: wrap the code of interest with preconditions and postconditions, then prove that postconditions are met
- Clarke, Emerson et Sifakis 1974: model checking
- Cousot(s) 1977: Abstract Interpretation

Properties of Interest

- run time errors: overflow, division by zero, square root of negatives, etc.
- robustness and stability of algorithms: linear and non linear recursive schemes, filters, etc.

Formal Verification Approaches

Formal Verification Approaches

- Hoare 1969: wrap the code of interest with preconditions and postconditions, then prove that postconditions are met
- Clarke, Emerson et Sifakis 1974: model checking
- Cousot(s) 1977: Abstract Interpretation

Properties of Interest

- run time errors: overflow, division by zero, square root of negatives, etc.
- robustness and stability of algorithms: linear and non linear recursive schemes, filters, etc.

Formal Verification Approaches

Formal Verification Approaches

- Hoare 1969: wrap the code of interest with preconditions and postconditions, then prove that postconditions are met
- Clarke, Emerson et Sifakis 1974: model checking
- Cousot(s) 1977: Abstract Interpretation

Properties of Interest

- run time errors: overflow, division by zero, square root of negatives, etc.
- robustness and stability of algorithms: linear and non linear recursive schemes, filters, etc.

Formal Verification Approaches

Formal Verification Approaches

- Hoare 1969: wrap the code of interest with preconditions and postconditions, then prove that postconditions are met
- Clarke, Emerson et Sifakis 1974: model checking
- Cousot(s) 1977: Abstract Interpretation

Properties of Interest

- run time errors: overflow, division by zero, square root of negatives, etc.
- robustness and stability of algorithms: linear and non linear recursive schemes, filters, etc.

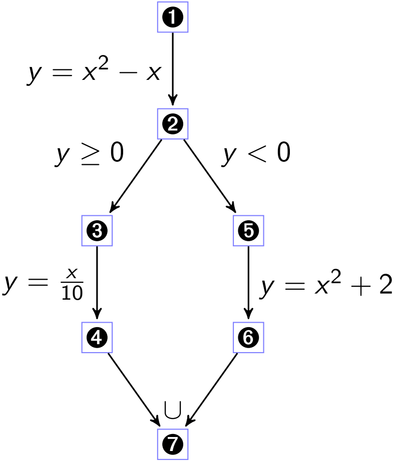
Abstract Interpretation, an overview

- Program semantics formalized as a fixpoint of a monotonic operator in a complete partially ordered set (exemplified later),
- Fully automated,
- Industrial tools exists : Polyspace Verifier (MathWorks), Astrée (ENS/ABSINT), Fluctuat (CEA), aIT (ABSINT), F-Soft (Nec Labs) ...

Challenge

find the *suitable* abstract domain for the properties of interest.

Equations System (collecting semantic)



$$\begin{cases} \mathcal{X}_1 = [\![\mathcal{V} \to \mathbb{I}]\!]^\flat \\ \mathcal{X}_2 = [\![y \leftarrow x^2 - x]\!]^\flat(\mathcal{X}_1) \\ \mathcal{X}_3 = [\![y \geq 0]\!]^\flat(\mathcal{X}_2) \\ \mathcal{X}_4 = [\![y \leftarrow \frac{x}{10}]\!]^\flat(\mathcal{X}_3) \\ \mathcal{X}_5 = [\![y < 0]\!]^\flat(\mathcal{X}_2) \\ \mathcal{X}_6 = [\![y \leftarrow x^2 + 2]\!]^\flat(\mathcal{X}_5) \\ \mathcal{X}_7 = \mathcal{X}_6 \cup \mathcal{X}_4 \end{cases}$$

Solving the equations system

- $D = (\wp(\mathcal{V} \to \mathbb{I}), \subseteq, \cup, \cap, \emptyset, (\mathcal{V} \to \mathbb{I}))$ is a **complete lattice**
- each operator $\mathcal{X} \mapsto \mathcal{F}(\mathcal{X})$ is monotonic

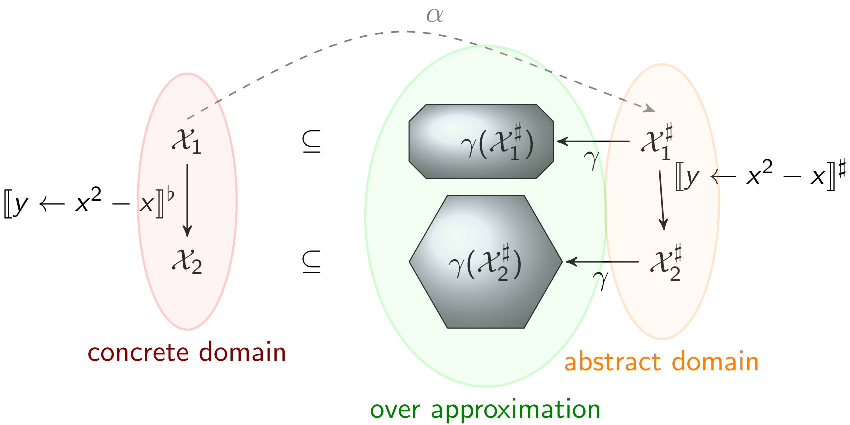
➜ **Tarski Theorem** ensures the existence of a least fixpoint for $\mathcal{F}$

➜ **Kleene Iteration Technique** reaches the least fixpoint

Issues

- $\wp(\mathcal{V} \to \mathbb{I})$ is non representable in finite memory,
- $[\![.]\!]^\flat$ are non computable,
- Iterations over the lattice may be transfinite.

Concretisation-Based Abstract Interpretation



Building an abstract domain

- lattice-like structure:
  - abstract objects
  - order relation (preorder over abstract objects)
  - monotonic concretisation function (γ)
- Transfer Functions
  - evaluation of arithmetic expressions ($[\![x^2 - x]\!]^\sharp$)
  - assignment ($\mathcal{X}_2 = [\![y \leftarrow x^2 - x]\!]^\sharp(\mathcal{X}_1)$)
  - upper bound (join) ($\mathcal{X}_7 = \mathcal{X}_6 \cup \mathcal{X}_4$)
  - over-approximation of lower bounds (meet) ($\mathcal{X}_3 = [\![y \geq 0]\!]^\sharp \mathcal{X}_2 =$ "$\mathcal{X}_3 = \mathcal{X}_2 \cap [\![y \geq 0]\!]^\sharp \top^\sharp$")
- Convergence acceleration (widening)

Outlines

Affine Sets

introduction

$$\hat{X} = \begin{cases} \hat{x}_1 = \alpha_0^{x_1} + \sum_{i=1}^{n} \alpha_i^{x_1} \epsilon_i \\ \hat{x}_2 = \alpha_0^{x_2} + \sum_{i=1}^{n} \alpha_i^{x_2} \epsilon_i \\ \hat{x}_3 = \dots \end{cases}, \quad (\epsilon_1, \dots, \epsilon_n) \in [-1, 1]^n$$

$$\hat{X} = \underbrace{\begin{pmatrix} \alpha_0^{x} & \cdots & \alpha_n^{x} \\ \vdots & & \vdots \\ \alpha_0^{x_p} & \cdots & \alpha_n^{x_p} \end{pmatrix}}_{C^X} \times \underbrace{\begin{pmatrix} \epsilon_0 \\ \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix}}_{\varepsilon^*}, \varepsilon \in 1 \times [-1, 1]^n$$

Geometric Concretisation: Zonotope

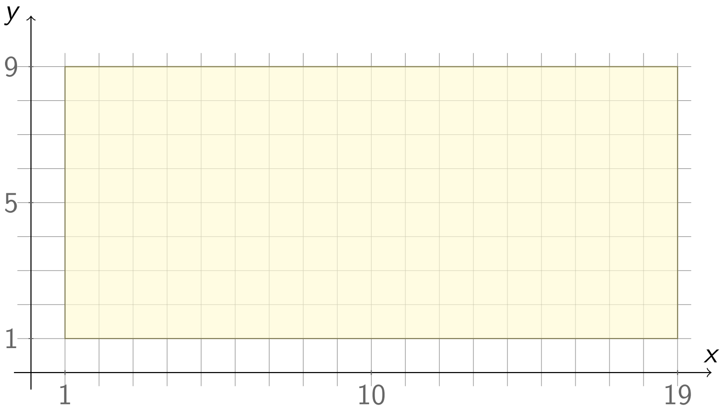
Minkowski sum of a set of segments

$$
\begin{aligned}
\hat{x} &= 10 - 4\epsilon_1 + 2\epsilon_3 + 3\epsilon_4 \\
\hat{y} &= 5 \;\; - 2\epsilon_1 + 1\epsilon_2 - 1\epsilon_4
\end{aligned}
$$

Geometric Concretisation: Zonotope

Minkowski sum of a set of segments

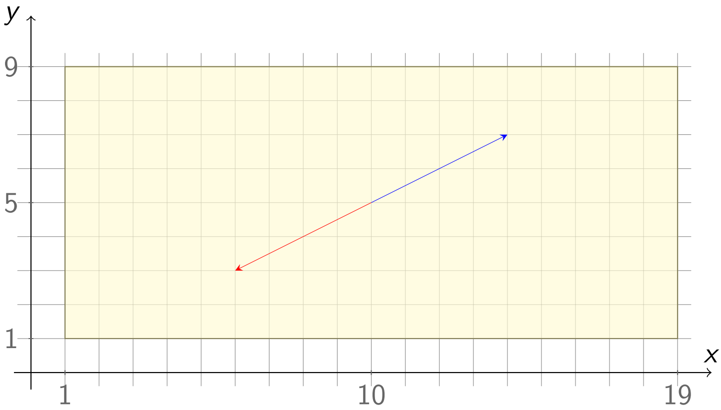
$$\begin{aligned} \hat{x} &= 10 - 4\epsilon_1 + 2\epsilon_3 + 3\epsilon_4 \\ \hat{y} &= 5 - 2\epsilon_1 + 1\epsilon_2 - 1\epsilon_4 \end{aligned} \longrightarrow \begin{aligned} x &\in [1, 19] \\ y &\in [1, 9] \end{aligned}$$



Geometric Concretisation: Zonotope

Minkowski sum of a set of segments

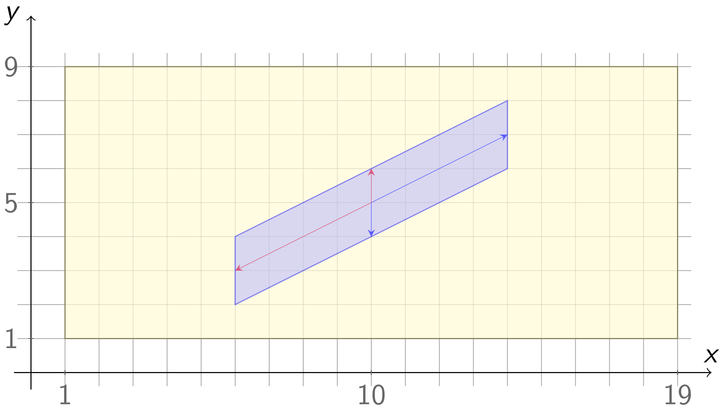
$$\begin{aligned}\hat{x} &= 10 - 4\epsilon_1 \quad\quad + 2\epsilon_3 + 3\epsilon_4 \\ \hat{y} &= 5 \;\; - 2\epsilon_1 + 1\epsilon_2 \quad\quad - 1\epsilon_4\end{aligned} \longrightarrow \begin{aligned} x &\in [1, 19] \\ y &\in [1, 9]\end{aligned}$$



Geometric Concretisation: Zonotope

Minkowski sum of a set of segments

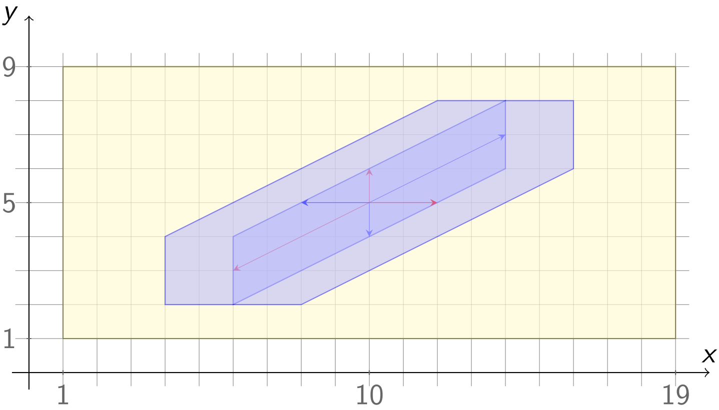
$$\begin{aligned}\hat{x} &= 10 - 4\epsilon_1 + 2\epsilon_3 + 3\epsilon_4 \\ \hat{y} &= 5 - 2\epsilon_1 + 1\epsilon_2 - 1\epsilon_4\end{aligned} \longrightarrow \begin{aligned} x &\in [1, 19] \\ y &\in [1, 9]\end{aligned}$$



Geometric Concretisation: Zonotope

Minkowski sum of a set of segments

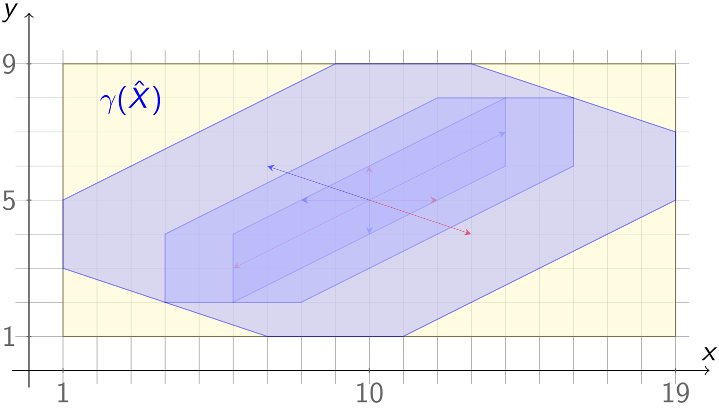
$$\begin{aligned} \hat{x} &= 10 - 4\epsilon_1 + 2\epsilon_3 + 3\epsilon_4 \\ \hat{y} &= 5 - 2\epsilon_1 + 1\epsilon_2 - 1\epsilon_4 \end{aligned} \longrightarrow \begin{aligned} x &\in [1, 19] \\ y &\in [1, 9] \end{aligned}$$



Geometric Concretisation: Zonotope

Minkowski sum of a set of segments

$$\begin{aligned} \hat{x} &= 10 - 4\epsilon_1 + 2\epsilon_3 + 3\epsilon_4 \\ \hat{y} &= 5 \;\; - 2\epsilon_1 + 1\epsilon_2 - 1\epsilon_4 \end{aligned} \longrightarrow \begin{aligned} x &\in [1, 19] \\ y &\in [1, 9] \end{aligned}$$



Perturbed Affine Sets

The Affine Sets are extended with **Perturbation** terms to handle the **non linear** operations: multiplication, join, etc.

$$\hat{X} = \begin{cases} \hat{x} = \alpha_0^x + \sum_{i=1}^n \alpha_i^x \epsilon_i + \sum_{j=1}^m \beta_j^x \eta_j^X \\ \hat{y} = \alpha_0^y + \sum_{i=1}^n \alpha_i^y \epsilon_i + \sum_{j=1}^m \beta_j^y \eta_j^X \\ \hat{z} = \dots \end{cases}$$

$$\begin{aligned} \hat{X} = C^X \varepsilon + P^X \eta^X, & \\ \varepsilon &= (\epsilon_0, \dots, \epsilon_n) \in 1 \times [-1,1]^n \\ \eta^X &= (\eta_1^X, \dots, \eta_m^X) \in [-1,1]^m \end{aligned}$$

- **p** numerical variables
- **n** input noise symbols
- **m** perturbation noise symbols

Functional Order Relation over Perturbed Affine Sets

Intuition

Geometrical inclusion of the concretisation of the vector $\hat{X}$ augmented by the input noise symbols ε.

$$\hat{X} \leq_1 \hat{Y}$$

$$\overset{\Delta}{\Longleftrightarrow}$$

$$\left\{\gamma(\hat{X},\varepsilon) \mid \hat{X} = C^X\varepsilon + P^X\eta^X\right\} \subseteq \left\{\gamma(\hat{Y},\varepsilon) \mid \hat{Y} = C^Y\varepsilon + P^Y\eta^Y\right\}$$

Functional Order Relation, Equivalent Formulations

Inclusion of sets of functions

$$\forall \varepsilon \in 1 \times [-1,1]^n, \forall \eta^X \in [-1,1]^m, \exists \eta^Y \in [-1,1]^m :$$
$$C^X \varepsilon + P^X \eta^X = C^Y \varepsilon + P^Y \eta^Y \ .$$

Sets Inclusion

$$(C^X - C^Y)\Phi_\varepsilon + P^X \Phi_\eta^X \subseteq P^Y \Phi_\eta^Y, \begin{cases} \Phi_\varepsilon = 1 \times [-1,1]^n \\ \Phi_\eta^X = \Phi_\eta^Y = [-1,1]^m \end{cases}$$

Support Function Inequality

$$\forall t \in \mathbb{R}^p, \ \sup_{\varepsilon \in \Phi_\varepsilon} |\langle (C^X - C^Y)\varepsilon, t\rangle| \leq \sup_{\eta^Y \in \Phi_\eta^Y} |\langle P^Y \eta^Y, t\rangle| - \sup_{\eta^X \in \Phi_\eta^X} |\langle P^X \eta^X, t\rangle|$$

From Sets Inclusion to Functions Inequality

The support function

Let C be a non empty convex set of $\mathbb{R}^p$, then

$$\delta(t \mid C) \stackrel{\text{def}}{=} \sup\left\{\langle t, x\rangle \mid x \in C\right\},$$

where $\langle\cdot,\cdot\rangle$ denotes the usual scalar product over $\mathbb{R}^p$.

Proposition

Let, C_1 and C_2 be non empty convex sets, then

$$C_1 \subseteq C_2 \iff \forall t \in \mathbb{R}^p, \delta(t \mid C_1) \leq \delta(t \mid C_2)$$

Support Function Inequality: Perturbed Affine Sets

Perturbed Affine Sets

$$\forall t \in \mathbb{R}^p,\ \sup_{\varepsilon \in \Phi_\varepsilon} |\langle (C^X - C^Y)\varepsilon, t\rangle| \leq \sup_{\eta^Y \in \Phi_\eta^Y} |\langle P^Y \eta^Y, t\rangle| - \sup_{\eta^X \in \Phi_\eta^X} |\langle P^X \eta^X, t\rangle|$$

Norm L_1 Formulation

$$\forall t \in \mathbb{R}^p,\ \|(C^X - C^Y)^* t\|_1 \leq \|P^{Y^*} t\|_1 - \|P^{X^*} t\|_1$$

where $x \in \mathbb{R}^n$, $\|x\|_1 \stackrel{\text{def}}{=} \sum_{i=1}^n |x_i|$.

Support Function Inequality: Perturbed Affine Sets

Perturbed Affine Sets

$$\forall t \in \mathbb{R}^p,\ \sup_{\varepsilon \in \Phi_\varepsilon} |\langle (C^X - C^Y)\varepsilon, t\rangle| \leq \sup_{\eta^Y \in \Phi_\eta^Y} |\langle P^Y \eta^Y, t\rangle| - \sup_{\eta^X \in \Phi_\eta^X} |\langle P^X \eta^X, t\rangle|$$

Norm L_1 Formulation

$$\forall t \in \mathbb{R}^p,\ \|(C^X - C^Y)^* t\|_1 \leq \|P^{Y^*} t\|_1 - \|P^{X^*} t\|_1$$

where $x \in \mathbb{R}^n, \|x\|_1 \stackrel{\text{def}}{=} \sum_{i=1}^n |x_i|$.

Arithmetic Operations on Perturbed Affine Forms

Linear Operations

Closed under affine transformations

$$\hat{x} \pm \hat{y} \stackrel{\text{def}}{=} \sum_{i=0}^{n} (\alpha_i^x \pm \alpha_i^y)\epsilon_i + \sum_{j=1}^{m} (\beta_j^x \pm \beta_j^y)\eta_j$$

$$\lambda.\hat{x} \stackrel{\text{def}}{=} \sum_{i=0}^{n} (\lambda\alpha_i^x)\epsilon_i + \sum_{j=1}^{m} (\lambda\beta_j^x)\eta_j$$

Proposition

The assignment of linear expression is monotonic.

Arithmetic Operations on Perturbed Affine Forms

Non Linear Unary Operations

For non linear (unary) operations (3 steps)

1. linearize using first order **Taylor development**
2. **bound** the non linear term
3. rewrite the interval as an affine form using a **fresh** noise symbol

Square root example

- $x \in [3, 5] : \hat{x} = 4 + \epsilon_1$
- $\hat{y} = \sqrt{\hat{x}} = 2 + 0.25\epsilon_1 + 0.024\epsilon_f$
- $\gamma(\hat{y}) = [1.726, 2.274] \supseteq [1.732, 2.236]$

Arithmetic Operations on Perturbed Affine Forms

Multiplication

Multiplication operation

$$\hat{x} \times \hat{y} = \alpha_0^x \alpha_0^y + \sum_{i=1}^{n} (\alpha_i^x \alpha_0^y + \alpha_i^y \alpha_0^x)\epsilon_i + \sum_{j=1}^{m} (\beta_j^x \alpha_0^y + \beta_j^y \alpha_0^x)\eta_j + \ell(\hat{x}, \hat{y})\eta_f \quad .$$

Two methods to bound the quadratic non linear term

1. Straightforward method: interval arithmetic
   - **rough** approximation but **efficient** computation
2. SemiDefinite Programming Technique
   - more **accurate** but more **expensive** on time

Comparative example

```
#My Simple Program
x = [0,2];
y = x + [0,2];
z = x*y;
t = z - 2*x - y;
```

Abstract computations

1. $\hat{x} = 1 + \epsilon_1$
2. $\hat{y} = 2 + \epsilon_1 + \epsilon_2$
3. $\hat{z} = 2.875 + 3\epsilon_1 + \epsilon_2 + 1.125\eta_1$
4. $\hat{t} = -1.125 + 1.125\eta_1$

Comparative example

```
#My Simple Program
x = [0,2];❶
y = x + [0,2];
z = x*y;
t = z - 2*x - y;
```

Abstract computations

❶ $\hat{x} = 1 + \epsilon_1$

❷ $\hat{y} = 2 + \epsilon_1 + \epsilon_2$

❸ $\hat{z} = 2.875 + 3\epsilon_1 + \epsilon_2 + 1.125\eta_1$

❹ $\hat{t} = -1.125 + 1.125\eta_1$

Comparative example

```
#My Simple Program
x = [0,2];❶
y = x + [0,2];❷
z = x*y;
t = z - 2*x - y;
```

Abstract computations

❶ $\hat{x} = 1 + \epsilon_1$

❷ $\hat{y} = 1 + \epsilon_1 + (1 + \epsilon_2) = 2 + \epsilon_1 + \epsilon_2$

❸ $\hat{z} = 2.875 + 3\epsilon_1 + \epsilon_2 + 1.125\eta_1$

❹ $\hat{t} = -1.125 + 1.125\eta_1$

Comparative example

```
#My Simple Program
x = [0,2];❶
y = x + [0,2];❷
z = x*y;❸
t = z - 2*x - y;
```

Abstract computations

❶ $\hat{x} = 1 + \epsilon_1$

❷ $\hat{y} = 2 + \epsilon_1 + \epsilon_2$

❸ $\hat{z} = 2.875 + 3\epsilon_1 + \epsilon_2 + 1.125\eta_1$

❹ $\hat{t} = -1.125 + 1.125\eta_1$

Comparative example

```
#My Simple Program
x = [0,2];❶
y = x + [0,2];❷
z = x*y;❸
t = z - 2*x - y;❹
```

Abstract computations

❶ $\hat{x} = 1 + \epsilon_1$

❷ $\hat{y} = 2 + \epsilon_1 + \epsilon_2$

❸ $\hat{z} = 2.875 + 3\epsilon_1 + \epsilon_2 + 1.125\eta_1$

❹ $\hat{t} = -1.125 + 1.125\eta_1$

Comparative example

```
#My Simple Program
x = [0,2];❶
y = x + [0,2];❷
z = x*y;❸
t = z - 2*x - y;❹
```

Abstract computations

❶ $\hat{x} = 1 + \epsilon_1$

❷ $\hat{y} = 2 + \epsilon_1 + \epsilon_2$

❸ $\hat{z} = 2.875 + 3\epsilon_1 + \epsilon_2 + 1.125\eta_1$

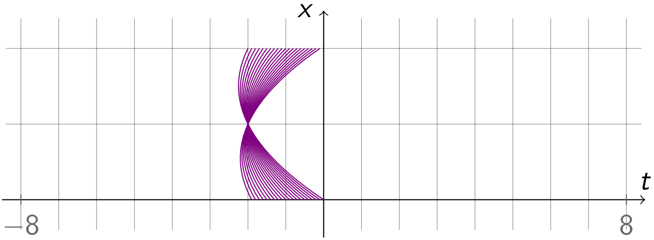
❹ $\hat{t} = -1.125 + 1.125\eta_1$

- (x,t) invariant in control point ❹

$$\text{Polyhedra} \left\{ \begin{array}{rl} t + x & \leq 4 \\ -t - 3x & \leq 2 \\ x & \leq 2 \\ -x & \leq 0 \end{array} \right. ;\ \text{Octagons} \left\{ \begin{array}{rl} t - x & \leq 8 \\ t + x & \leq 8 \\ -t - x & \leq 6 \\ -t + x & \leq 10 \\ t & \leq 8 \\ -t & \leq 8 \\ x & \leq 2 \\ -x & \leq 0 \end{array} \right.$$

Comparative example

Concretisation of abstract set (❹) projected onto (t,x) plane

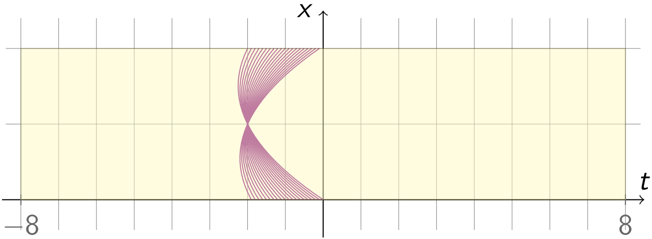


Interval concretisation of variable t

- Octagons $t \in [-8, 8]$
- Polyhedra $t \in [-8, 4]$
- PAS $t \in [-2.25, 0]$ $(\hat{t} = -1.125 + 1.125\eta_1)$

Comparative example

Concretisation of abstract set (❹) projected onto (t,x) plane

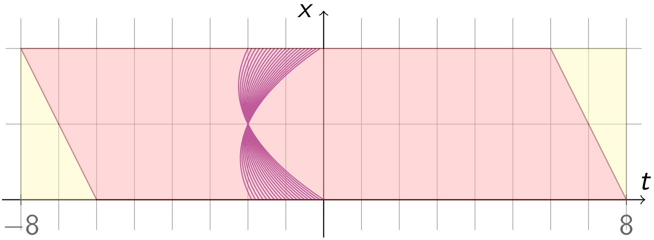


Interval concretisation of variable t

- Octagons $t \in [-8, 8]$
- Polyhedra $t \in [-8, 4]$
- PAS $t \in [-2.25, 0]$ $(\hat{t} = -1.125 + 1.125\eta_1)$

Comparative example

Concretisation of abstract set (❹) projected onto (t,x) plane

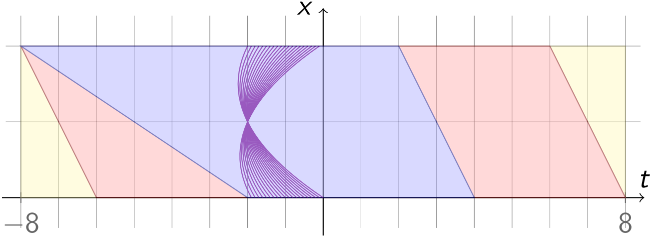


Interval concretisation of variable t

- Octagons $t \in [-8, 8]$
- Polyhedra $t \in [-8, 4]$
- PAS $t \in [-2.25, 0]$ $(\hat{t} = -1.125 + 1.125\eta_1)$

Comparative example

Concretisation of abstract set (❹) projected onto (t,x) plane

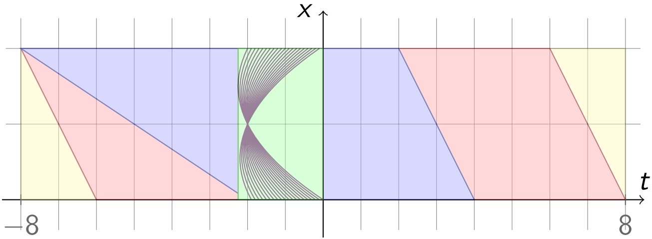


Interval concretisation of variable t

- Octagons $t \in [-8, 8]$
- Polyhedra $t \in [-8, 4]$
- PAS $t \in [-2.25, 0]$ $(\hat{t} = -1.125 + 1.125\eta_1)$

Comparative example

Concretisation of abstract set (❹) projected onto (t,x) plane



Interval concretisation of variable t

- Octagons $t \in [-8, 8]$
- Polyhedra $t \in [-8, 4]$
- PAS $t \in [-2.25, 0]$ $(\hat{t} = -1.125 + 1.125\eta_1)$

Join over Perturbed Affine Sets

- we do not have a supremum in general,
- many "minimal" upper bounds exist,
- the best (if many) minimal upper bound depends on the future evaluations
- computing a minimal enclosing zonotope of two given zonotopes is a *hard problem*

Minimal Upper Bound with respect to $\preceq$ / $\sim$

Z is a minimal upper bound of X and Y if and only if

- **upper bound**: $X \preceq Z$ and $Y \preceq Z$, and
- **minimal**: for all W upper bound of X and Y, $Z \preceq W \implies Z \sim W$

Perturbed Affine Forms

One Dimensional Affine Set

A Perturbed Affine Form is nothing but 1-dim a Perturbed Affine Sets (with one perturbation noise symbol).

$$\hat{x} = \alpha_0^x + \sum_{i=1}^{n} \alpha_i^x \epsilon_i + \beta^x \eta_u^x$$

and therefore

$$\hat{x} \leq_1 \hat{y} \iff \|\alpha^x - \alpha^y\|_1 \leq |\beta^y| - |\beta^x|$$

Minimal Upper Bound of Two Perturbed Affine Forms

Proposition

The following $\hat{z}$ is a minimal upper bound of $\hat{x}$ and $\hat{y}$ (if $\hat{x}$ and $\hat{y}$ are non comparable) whose **interval concretisation is the union of the interval concretizations** of $\hat{x}$ and $\hat{y}$:

$$
\begin{aligned}
\alpha_0^z &= mid(\gamma(\hat{x}) \cup \gamma(\hat{y})) && \text{(central value of } \hat{z}\text{)} \\
\alpha_i^z &= \operatorname*{argmin}_{min(\alpha_i^x,\alpha_i^y) \le \alpha \le max(\alpha_i^x,\alpha_i^y)} (|\alpha|), \forall i \ge 1 && \text{(coeff. of } \epsilon_i\text{)} \\
\beta^z &= \sup(\gamma(\hat{x}) \cup \gamma(\hat{y})) - \alpha_0^z - \textstyle\sum_{i \ge 1} |\alpha_i^z| && \text{(coeff. of } \epsilon_U\text{)}
\end{aligned}
$$

where :

- $\gamma(\hat{x}) = [\alpha_0^x - \sum_{i=1}^n |\alpha_i^x|, \alpha_0^x + \sum_{i=1}^n |\alpha_i^x|]$,
- and $mid([a, b]) := \frac{1}{2}(a + b)$,
- and $\operatorname*{argmin}_{a \le x \le b}(|x|) := \{x \in [a, b], |x| \text{ is minimal } \}$.

Componentwise Relaxation

$$\hat{X} = \begin{pmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_p \end{pmatrix} \leq_1 \hat{Y} = \begin{pmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_p \end{pmatrix} \qquad \Rightarrow \qquad \forall i, 1 \leq i \leq p, \hat{x}_i \leq_1 \hat{y}_i$$

- Moreover, when the Perturbation Matrices P^Y and P^X are diagonal, the equivalence holds
- We have a **linear algorithm** to compute a **minimal upper bound** of two Perturbed Affine Forms.
- Idea: use componentwise minimal upper bounds computation to define an upper bound of $\hat{X}$ and $\hat{Y}$ in the general case.

Componentwise Relaxation

$$\hat{X} = \begin{pmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_p \end{pmatrix} \leq_1 \hat{Y} = \begin{pmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_p \end{pmatrix} \quad \begin{matrix} \Rightarrow \\ \not\Leftarrow \end{matrix} \quad \forall i, 1 \leq i \leq p, \hat{x}_i \leq_1 \hat{y}_i$$

- Moreover, when the Perturbation Matrices P^Y and P^X are diagonal, the equivalence holds
- We have a **linear algorithm** to compute a **minimal upper bound** of two Perturbed Affine Forms.
- Idea: use componentwise minimal upper bounds computation to define an upper bound of $\hat{X}$ and $\hat{Y}$ in the general case.

Componentwise Relaxation

$$\hat{X} = \begin{pmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_p \end{pmatrix} \leq_1 \hat{Y} = \begin{pmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_p \end{pmatrix} \quad \begin{array}{c} \Rightarrow \\ \not\Leftarrow \end{array} \quad \forall i, 1 \leq i \leq p, \hat{x}_i \leq_1 \hat{y}_i$$

- Moreover, when the Perturbation Matrices P^Y and P^X are diagonal, the equivalence holds
- We have a **linear algorithm** to compute a **minimal upper bound** of two Perturbed Affine Forms.
- Idea: use componentwise minimal upper bounds computation to define an upper bound of $\hat{X}$ and $\hat{Y}$ in the general case.

Componentwise Relaxation

$$\hat{X} = \begin{pmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_p \end{pmatrix} \leq_1 \hat{Y} = \begin{pmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_p \end{pmatrix} \quad \begin{matrix} \Rightarrow \\ \\ \not\Leftarrow \end{matrix} \quad \forall i, 1 \leq i \leq p, \hat{x}_i \leq_1 \hat{y}_i$$

- Moreover, when the Perturbation Matrices P^Y and P^X are diagonal, the equivalence holds
- We have a **linear algorithm** to compute a **minimal upper bound** of two Perturbed Affine Forms.
- Idea: use componentwise minimal upper bounds computation to define an upper bound of $\hat{X}$ and $\hat{Y}$ in the general case.

Componentwise Relaxation

$$\hat{X} = \begin{pmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_p \end{pmatrix} \leq_1 \hat{Y} = \begin{pmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_p \end{pmatrix} \quad \begin{matrix} \Rightarrow \\ \\ \not\Leftarrow \end{matrix} \quad \forall i, 1 \leq i \leq p, \hat{x}_i \leq_1 \hat{y}_i$$

- Moreover, when the Perturbation Matrices P^Y and P^X are diagonal, the equivalence holds
- We have a **linear algorithm** to compute a **minimal upper bound** of two Perturbed Affine Forms.
- Idea: use componentwise minimal upper bounds computation to define an upper bound of $\hat{X}$ and $\hat{Y}$ in the general case.

Join : Relaxing the Problem

We Over-approximate $\hat{X}$ and $\hat{Y}$

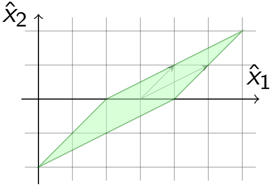
- $\hat{X} \leq_1 \hat{X}' = C^X \varepsilon + P^{X'} \eta^X$,
  - $P^{X'}$ is a **diagonal** matrix,
  - $P^{X'}_{i,i} = \delta(P^X e_i \mid \Phi^X_\eta)(= \|(\beta_1^{x_i}, \ldots, \beta_m^{x_i})\|_1)$.
- The over-approximation of $\hat{Y}$ by $\hat{Y}'$ is similar.

- We compute componentwisely a Minimal Upper Bound of $\hat{X}'$ and $\hat{Y}'$.

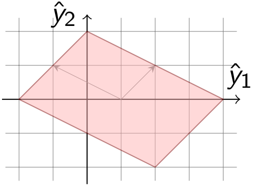
→ We get an upper bound of $\hat{X}$ and $\hat{Y}$.

Example: Join over Perturbed Affine Sets

$$\begin{pmatrix} \hat{x}_1 = 3 + \epsilon_1 + 2\epsilon_2 \\ \hat{x}_2 = 0 + \epsilon_1 + \epsilon_2 \end{pmatrix} \quad \cup \quad \begin{pmatrix} \hat{y}_1 = 1 - 2\epsilon_1 + \epsilon_2 \\ \hat{y}_2 = 0 + \epsilon_1 + \epsilon_2 \end{pmatrix}$$



$\cup$



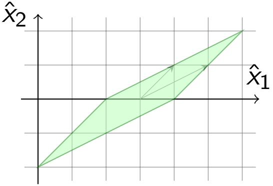
$$\begin{pmatrix} \hat{z}_1 = 2 + \epsilon_2 + 3\eta_u^z \\ \hat{z}_2 = \epsilon_1 + \epsilon_2 \end{pmatrix}$$

Properties

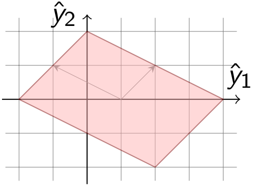
- $\gamma(\hat{z}_i) = \gamma(\hat{x}_i) \cup \gamma(\hat{y}_i)$
- Complexity : $\mathcal{O}(pn)$

Example: Join over Perturbed Affine Sets

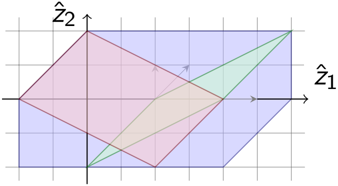
$$\begin{pmatrix} \hat{x}_1 = 3 + \epsilon_1 + 2\epsilon_2 \\ \hat{x}_2 = 0 + \epsilon_1 + \epsilon_2 \end{pmatrix} \cup \begin{pmatrix} \hat{y}_1 = 1 - 2\epsilon_1 + \epsilon_2 \\ \hat{y}_2 = 0 + \epsilon_1 + \epsilon_2 \end{pmatrix}$$



$\cup$



$$\begin{pmatrix} \hat{z}_1 = 2 + \epsilon_2 + 3\eta_u^z \\ \hat{z}_2 = \epsilon_1 + \epsilon_2 \end{pmatrix}$$



Properties

- $\gamma(\hat{z}_i) = \gamma(\hat{x}_i) \cup \gamma(\hat{y}_i)$
- Complexity : $\mathcal{O}(pn)$

Meet Operation

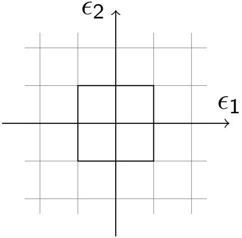
Issues

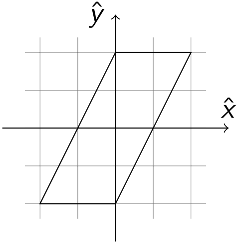
- Unlike the join, choosing a maximal lower bound is not sound,
- The set of Affine Forms is not a Riez space, that is $-((-\hat{x}) \cup (-\hat{y}))$ does not give a a maximal lower bound in general,
- The intersection of a hyperplane and a zonotope is not a zonotope in general,
- The meet of two non equal non perturbed affine forms ($\beta^x = \beta^y = 0$) is the bottom element,

↠ Tests are mainly ignored in Perturbed Affine Sets Abstract Domain, which is **sound** but too **pessimistic**. **Reduced product** with intervals is used to improve the precision.

Interpretation of Tests

Intuitions





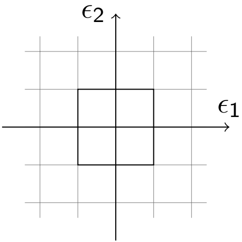
$$\forall i, \epsilon_i \in [-1, 1]$$

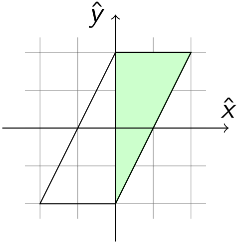
$$\begin{aligned} \hat{x} &= \epsilon_1 - \epsilon_2 \\ \hat{y} &= 2\epsilon_1 \\ 0 &\leq \hat{x} \end{aligned}$$

- The intersection is a **general polytope**
- **Propagate** the constraint on noise symbols

Interpretation of Tests

Intuitions





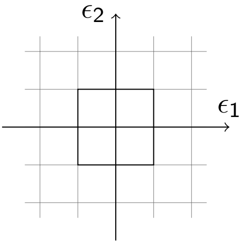
$\forall i, \epsilon_i \in [-1, 1]$

$$\begin{aligned} \hat{x} &= \epsilon_1 - \epsilon_2 \\ \hat{y} &= 2\epsilon_1 \\ 0 &\leq \hat{x} \end{aligned}$$

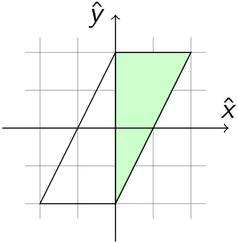
- The intersection is a **general polytope**
- **Propagate** the constraint on noise symbols

Interpretation of Tests

Intuitions



$$\forall i, \epsilon_i \in [-1, 1]$$

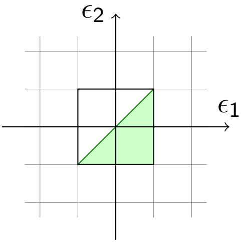


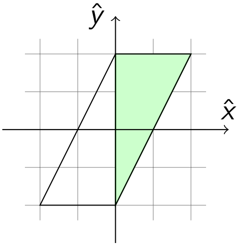
$$\begin{aligned} \hat{x} &= \epsilon_1 - \epsilon_2 \\ \hat{y} &= 2\epsilon_1 \\ 0 &\leq \hat{x} \end{aligned}$$

- The intersection is a **general polytope**
- **Propagate** the constraint on noise symbols

Interpretation of Tests

Intuitions





$\forall i, \epsilon_i \in [-1, 1]$

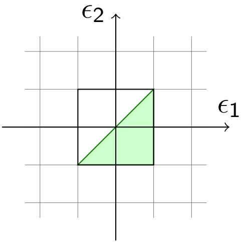
$\epsilon_2 \leq \epsilon_1$

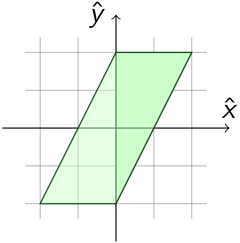
$$\begin{aligned} \hat{x} &= \epsilon_1 - \epsilon_2 \\ \hat{y} &= 2\epsilon_1 \\ 0 &\leq \hat{x} \end{aligned}$$

- The intersection is a **general polytope**
- **Propagate** the constraint on noise symbols

Interpretation of Tests

Intuitions





$\forall i, \epsilon_i \in [-1, 1]$

$\epsilon_2 \leq \epsilon_1$

$$\begin{aligned} \hat{x} &= \epsilon_1 - \epsilon_2 \\ \hat{y} &= 2\epsilon_1 \\ 0 &\leq \hat{x} \end{aligned}$$

- The intersection is a **general polytope**
- **Propagate** the constraint on noise symbols

Outlines

Constrained Affine Sets

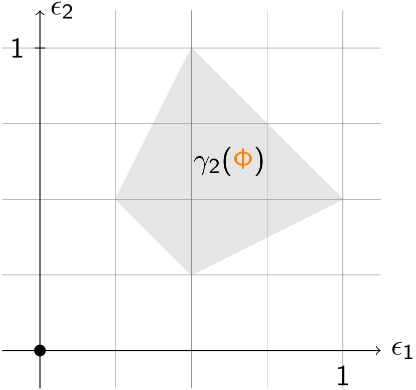
$$\hat{X} = \begin{cases} \hat{x} = \alpha_0^x + \sum_{i=1}^{n} \alpha_i^x \epsilon_i + \sum_{j=1}^{m} \beta_j^x \eta_j^X \\ \hat{y} = \alpha_0^y + \sum_{i=1}^{n} \alpha_i^y \epsilon_i + \sum_{j=1}^{m} \beta_j^y \eta_j^X \\ \hat{z} = \dots \end{cases}$$

$$\hat{X} = C^X \varepsilon + P^X \eta^X, (\varepsilon, \eta^X) \in \gamma_2(\Phi^X)$$

Φ^X is an element of another abstract domain $\mathcal{A}_2$ (boxes, octagons, polyhedra etc.) .

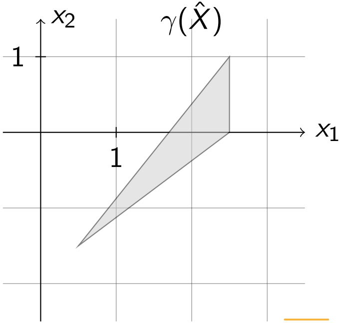
Geometric Concretisation

not a zonotope in general



$$\eta_u^{x_1}, \eta_u^{x_2} \in [-1, 1]$$

$$\hat{X} = \begin{cases} \hat{x}_1 = 1 + \epsilon_1 + \epsilon_2 + \eta_u^{x_1} \\ \hat{x}_2 = -1 + 2\epsilon_2 + \eta_u^{x_2} \end{cases}, \Phi$$



Order Relation over Constrained Affine Sets

$$\hat{X} = \begin{pmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_p \end{pmatrix}, \Phi^X \qquad \leq_{1\times 2} \qquad \hat{Y} = \begin{pmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_p \end{pmatrix}, \Phi^Y$$

$$\stackrel{\Delta}{\Longleftrightarrow}$$

$$\Phi^X \leq_2 \Phi^Y$$

$$\{\gamma(\hat{X}, \varepsilon) \mid \hat{X} = C^X\varepsilon + P^X\eta^X, (\varepsilon, \eta^X) \in \gamma_2(\Phi^X)\}$$
$$\subseteq$$
$$\{\gamma(\hat{Y}, \varepsilon) \mid \hat{X} = C^Y\varepsilon + P^Y\eta^Y, (\varepsilon, \eta^Y) \in \gamma_2(\Phi^Y)\}$$

Upper Bound of Two Constrained Affine Sets

- $(\hat{X}, \Phi^X) \leq_{1\times 2} (\hat{X}, \square\Phi^X)$,
- $\square\Phi^X$ defined as the smallest box that contains $\gamma_2(\Phi^X)$,
- Similarly for $(\hat{Y}, \Phi^Y)$ is over-approximated by $(\hat{Y}, \square\Phi^Y)$
- We then compute an upper bound of $(\hat{X}, \square\Phi^X)$ and $(\hat{Y}, \square\Phi^Y)$:
- using "Diagonal" Perturbation Sets (as seen in the non constrained case)

We therefore need to compute a minimal lower bound of two Constrained Affine Forms.

Upper Bound of Two Constrained Affine Sets

- $(\hat{X}, \Phi^X) \leq_{1\times 2} (\hat{X}, \square\Phi^X)$,
- $\square\Phi^X$ defined as the smallest box that contains $\gamma_2(\Phi^X)$,
- Similarly for $(\hat{Y}, \Phi^Y)$ is over-approximated by $(\hat{Y}, \square\Phi^Y)$
- We then compute an upper bound of $(\hat{X}, \square\Phi^X)$ and $(\hat{Y}, \square\Phi^Y)$:
- using "Diagonal" Perturbation Sets (as seen in the non constrained case)

We therefore need to compute a minimal lower bound of two Constrained Affine Forms.

Upper Bound of Two Constrained Affine Sets

- $(\hat{X}, \Phi^X) \leq_{1\times 2} (\hat{X}, \Box\Phi^X)$,
- $\Box\Phi^X$ defined as the smallest box that contains $\gamma_2(\Phi^X)$,
- Similarly for $(\hat{Y}, \Phi^Y)$ is over-approximated by $(\hat{Y}, \Box\Phi^Y)$
- We then compute an upper bound of $(\hat{X}, \Box\Phi^X)$ and $(\hat{Y}, \Box\Phi^Y)$:
- using "Diagonal" Perturbation Sets (as seen in the non constrained case)

We therefore need to compute a minimal lower bound of two Constrained Affine Forms.

Upper Bound of Two Constrained Affine Sets

- $(\hat{X}, \Phi^X) \leq_{1\times 2} (\hat{X}, \Box\Phi^X)$,
- $\Box\Phi^X$ defined as the smallest box that contains $\gamma_2(\Phi^X)$,
- Similarly for $(\hat{Y}, \Phi^Y)$ is over-approximated by $(\hat{Y}, \Box\Phi^Y)$
- We then compute an upper bound of $(\hat{X}, \Box\Phi^X)$ and $(\hat{Y}, \Box\Phi^Y)$:
- using "Diagonal" Perturbation Sets (as seen in the non constrained case)

We therefore need to compute a minimal lower bound of two Constrained Affine Forms.

Upper Bound of Two Constrained Affine Sets

- $(\hat{X}, \Phi^X) \leq_{1\times 2} (\hat{X}, \Box\Phi^X)$,
- $\Box\Phi^X$ defined as the smallest box that contains $\gamma_2(\Phi^X)$,
- Similarly for $(\hat{Y}, \Phi^Y)$ is over-approximated by $(\hat{Y}, \Box\Phi^Y)$
- We then compute an upper bound of $(\hat{X}, \Box\Phi^X)$ and $(\hat{Y}, \Box\Phi^Y)$:
- using "Diagonal" Perturbation Sets (as seen in the non constrained case)

We therefore need to compute a minimal lower bound of two Constrained Affine Forms.

Upper Bound of Two Constrained Affine Sets

- $(\hat{X}, \Phi^X) \leq_{1\times 2} (\hat{X}, \Box\Phi^X)$,
- $\Box\Phi^X$ defined as the smallest box that contains $\gamma_2(\Phi^X)$,
- Similarly for $(\hat{Y}, \Phi^Y)$ is over-approximated by $(\hat{Y}, \Box\Phi^Y)$
- We then compute an upper bound of $(\hat{X}, \Box\Phi^X)$ and $(\hat{Y}, \Box\Phi^Y)$:
- using "Diagonal" Perturbation Sets (as seen in the non constrained case)

We therefore need to compute a minimal lower bound of two Constrained Affine Forms.

Upper Bound of Two Constrained Affine Sets

- $(\hat{X}, \Phi^X) \leq_{1\times 2} (\hat{X}, \square\Phi^X)$,
- $\square\Phi^X$ defined as the smallest box that contains $\gamma_2(\Phi^X)$,
- Similarly for $(\hat{Y}, \Phi^Y)$ is over-approximated by $(\hat{Y}, \square\Phi^Y)$
- We then compute an upper bound of $(\hat{X}, \square\Phi^X)$ and $(\hat{Y}, \square\Phi^Y)$:
- using "Diagonal" Perturbation Sets (as seen in the non constrained case)

We therefore need to compute a minimal lower bound of two Constrained Affine Forms.

Order Relation over Constrained Affine Forms

$$\eta_u^x \in [-1, 1] \qquad\qquad \eta_u^y \in [-1, 1]$$

$$\hat{x} = \underbrace{\alpha_0^x + \sum_{i=1}^{n} \alpha_i^x \epsilon_i}_{x(\epsilon)} + \beta^x \eta_u^x \quad \leq_{1\times 2} \quad \hat{y} = \underbrace{\alpha_0^y + \sum_{i=1}^{n} \alpha_i^y \epsilon_i}_{y(\epsilon)} + \beta^y \eta_u^y$$

$$\stackrel{\Delta}{\Longleftrightarrow}$$

$$\sup_{\epsilon \in \square(\Phi_\epsilon^X)} |x(\epsilon) - y(\epsilon)| \leq |\beta^y| - |\beta^x|$$

Computing $\sup_{\epsilon \in \Box(\Phi_\epsilon)} |x(\epsilon) - y(\epsilon)|$

Computing the supremum over $\Box(\Phi_\epsilon^X)$

$$\sup_{\epsilon \in \Box(\Phi_\epsilon)} |x(\epsilon) - y(\epsilon)| = \delta\big(\alpha^x - \alpha^y \mid \mathrm{convex}(1 \times \Box(\Phi_\epsilon^X), -1 \times -\Box(\Phi_\epsilon^X))\big),$$

where $\alpha^x = (\alpha_0^x, \ldots, \alpha_n^x)$, and $\alpha^y = (\alpha_0^y, \ldots, \alpha_n^y)$.

A Particular Case: Non Constrained Case

- $\Box(\Phi_\epsilon^X) = [-1, 1]^n$
- $\mathrm{convex}(1 \times \Box(\Phi_\epsilon^X), -1 \times -\Box(\Phi_\epsilon^X)) = M^{X^*}\mathcal{B}^{n+1}$
- $\mathrm{convex}(1 \times \Box(\Phi_\epsilon^X), -1 \times -\Box(\Phi_\epsilon^X)) = [-1, 1]^{n+1}$
- $\delta(\alpha^x - \alpha^y \mid [-1, 1]^{n+1}) = \|\alpha^x - \alpha^y\|_1$

Indeed we have already seen in the **non constrained** case that

$$\sup_{\epsilon} |x(\epsilon) - y(\epsilon)| = \|\alpha^x - \alpha^y\|_1 \quad .$$

Computing $\sup_{\epsilon\in\Box(\Phi_\epsilon)}|x(\epsilon) - y(\epsilon)|$

Computing the supremum over $\Box(\Phi_\epsilon^X)$

$$\sup_{\epsilon\in\Box(\Phi_\epsilon)} |x(\epsilon) - y(\epsilon)| = \delta\left(\alpha^x - \alpha^y \mid \text{convex}(1 \times \Box(\Phi_\epsilon^X), -1 \times -\Box(\Phi_\epsilon^X))\right),$$

where $\alpha^x = (\alpha_0^x, \ldots, \alpha_n^x)$, and $\alpha^y = (\alpha_0^y, \ldots, \alpha_n^y)$.

A Particular Case: Non Constrained Case

- $\Box(\Phi_\epsilon^X) = [-1, 1]^n$
- $\text{convex}(1 \times \Box(\Phi_\epsilon^X), -1 \times -\Box(\Phi_\epsilon^X)) = M^{X^*}\mathcal{B}^{n+1}$
- $\text{convex}(1 \times \Box(\Phi_\epsilon^X), -1 \times -\Box(\Phi_\epsilon^X)) = [-1, 1]^{n+1}$
- $\delta(\alpha^x - \alpha^y \mid [-1, 1]^{n+1}) = \|\alpha^x - \alpha^y\|_1$

Indeed we have already seen in the **non constrained** case that

$$\sup_{\epsilon}|x(\epsilon) - y(\epsilon)| = \|\alpha^x - \alpha^y\|_1 \quad .$$

Computing $\sup_{\epsilon \in \square(\Phi_\epsilon)} |x(\epsilon) - y(\epsilon)|$

Computing the supremum over $\square(\Phi_\epsilon^X)$

$$\sup_{\epsilon \in \square(\Phi_\epsilon)} |x(\epsilon) - y(\epsilon)| = \delta\big(\alpha^x - \alpha^y \mid \text{convex}(1 \times \square(\Phi_\epsilon^X), -1 \times -\square(\Phi_\epsilon^X))\big),$$

where $\alpha^x = (\alpha_0^x, \ldots, \alpha_n^x)$, and $\alpha^y = (\alpha_0^y, \ldots, \alpha_n^y)$.

A Particular Case: Non Constrained Case

- $\square(\Phi_\epsilon^X) = [-1, 1]^n$
- $\text{convex}(1 \times \square(\Phi_\epsilon^X), -1 \times -\square(\Phi_\epsilon^X)) = M^{X^*}\mathcal{B}^{n+1}$
- $\text{convex}(1 \times \square(\Phi_\epsilon^X), -1 \times -\square(\Phi_\epsilon^X)) = [-1, 1]^{n+1}$
- $\delta(\alpha^x - \alpha^y \mid [-1, 1]^{n+1}) = \|\alpha^x - \alpha^y\|_1$

Indeed we have already seen in the **non constrained** case that

$$\sup_{\epsilon} |x(\epsilon) - y(\epsilon)| = \|\alpha^x - \alpha^y\|_1 \quad .$$

Computing $\sup_{\epsilon \in \Box(\Phi_\epsilon)} |x(\epsilon) - y(\epsilon)|$

Computing the supremum over $\Box(\Phi_\epsilon^X)$

$$\sup_{\epsilon \in \Box(\Phi_\epsilon)} |x(\epsilon) - y(\epsilon)| = \delta\left(\alpha^x - \alpha^y \mid \mathrm{convex}(1 \times \Box(\Phi_\epsilon^X), -1 \times -\Box(\Phi_\epsilon^X))\right),$$

where $\alpha^x = (\alpha_0^x, \ldots, \alpha_n^x)$, and $\alpha^y = (\alpha_0^y, \ldots, \alpha_n^y)$.

A Particular Case: Non Constrained Case

- $\Box(\Phi_\epsilon^X) = [-1, 1]^n$
- $\mathrm{convex}(1 \times \Box(\Phi_\epsilon^X), -1 \times -\Box(\Phi_\epsilon^X)) = M^{X^*}\mathcal{B}^{n+1}$
- $\mathrm{convex}(1 \times \Box(\Phi_\epsilon^X), -1 \times -\Box(\Phi_\epsilon^X)) = [-1, 1]^{n+1}$
- $\delta(\alpha^x - \alpha^y \mid [-1, 1]^{n+1}) = \|\alpha^x - \alpha^y\|_1$

Indeed we have already seen in the **non constrained** case that

$$\sup_{\epsilon} |x(\epsilon) - y(\epsilon)| = \|\alpha^x - \alpha^y\|_1 \quad .$$

Computing $\sup_{\epsilon \in \square(\Phi_\epsilon)} |x(\epsilon) - y(\epsilon)|$

Computing the supremum over $\square(\Phi_\epsilon^X)$

$$\sup_{\epsilon \in \square(\Phi_\epsilon)} |x(\epsilon) - y(\epsilon)| = \delta\left(\alpha^x - \alpha^y \mid \text{convex}(1 \times \square(\Phi_\epsilon^X), -1 \times -\square(\Phi_\epsilon^X))\right),$$

where $\alpha^x = (\alpha_0^x, \ldots, \alpha_n^x)$, and $\alpha^y = (\alpha_0^y, \ldots, \alpha_n^y)$.

A Particular Case: Non Constrained Case

- $\square(\Phi_\epsilon^X) = [-1, 1]^n$
- $\text{convex}(1 \times \square(\Phi_\epsilon^X), -1 \times -\square(\Phi_\epsilon^X)) = M^{X^*}\mathcal{B}^{n+1}$
- $\text{convex}(1 \times \square(\Phi_\epsilon^X), -1 \times -\square(\Phi_\epsilon^X)) = [-1, 1]^{n+1}$
- $\delta(\alpha^x - \alpha^y \mid [-1, 1]^{n+1}) = \|\alpha^x - \alpha^y\|_1$

Indeed we have already seen in the **non constrained** case that

$$\sup_{\epsilon} |x(\epsilon) - y(\epsilon)| = \|\alpha^x - \alpha^y\|_1 \quad .$$

Computing $\sup_{\epsilon \in \square(\Phi_\epsilon)} |x(\epsilon) - y(\epsilon)|$

Computing the supremum over $\square(\Phi_\epsilon^X)$

$$\sup_{\epsilon \in \square(\Phi_\epsilon)} |x(\epsilon) - y(\epsilon)| = \delta\left(\alpha^x - \alpha^y \mid \mathrm{convex}(1 \times \square(\Phi_\epsilon^X), -1 \times -\square(\Phi_\epsilon^X))\right),$$

where $\alpha^x = (\alpha_0^x, \ldots, \alpha_n^x)$, and $\alpha^y = (\alpha_0^y, \ldots, \alpha_n^y)$.

A Particular Case: Non Constrained Case

- $\square(\Phi_\epsilon^X) = [-1, 1]^n$
- $\mathrm{convex}(1 \times \square(\Phi_\epsilon^X), -1 \times -\square(\Phi_\epsilon^X)) = M^{X^*}\mathcal{B}^{n+1}$
- $\mathrm{convex}(1 \times \square(\Phi_\epsilon^X), -1 \times -\square(\Phi_\epsilon^X)) = [-1, 1]^{n+1}$
- $\delta(\alpha^x - \alpha^y \mid [-1, 1]^{n+1}) = \|\alpha^x - \alpha^y\|_1$

Indeed we have already seen in the **non constrained** case that

$$\sup_{\epsilon} |x(\epsilon) - y(\epsilon)| = \|\alpha^x - \alpha^y\|_1 \ .$$

Join Operation

Two (Distinct in general) Sufficient Conditions for Minimality

① $|\beta^z|$ is minimal over all upper bounds

② $|\beta^z|$ is minimal over all upper bounds which minimize the interval concretisation of $\hat{z}$

give in general different minimal upper bound

$$\begin{cases} \Phi^a = [-1, 0] \times [0, 0.5] \\ \hat{a} = 1 - \epsilon_1 + 2\epsilon_2 \\ \gamma(\hat{a}) = [1, 3] \end{cases} \quad \begin{cases} \Phi^b = [-0.5, 0.5] \times [0, 1] \\ \hat{b} = 2 + \epsilon_1 + \epsilon_2 \\ \gamma(\hat{b}) = [1.5, 3.5] \end{cases}$$

Two (non trivial) Minimal Upper Bounds:

$$② \begin{cases} \Phi^c = [-1, 0.5] \times [0, 1] \\ \hat{c} = 1.75 + \epsilon_2 + 0.75\eta_u^c \\ \gamma(\hat{c}) = [1, 3.5] = [1, 3] \cup [1.5, 3.5] \end{cases} \quad ① \begin{cases} \Phi^d = [-1, 0.5] \times [0, 1] \\ \hat{d} = 1.7 + 0.2\epsilon_1 + 1.6\epsilon_2 + 0.7\eta_u^d \\ \gamma(\hat{d}) = [0.8, 4.1] \supseteq [1, 3.5] \end{cases}$$

Which α^z minimizes $|\beta^z|$?

$\hat{z}$ is an upper bound :

- $\hat{x} \leq_{1\times 2} \hat{z} \iff \delta(M^X(\alpha^z - \alpha^x) \mid \mathcal{B}^{n+1}) \leq |\beta^z| - |\beta^x|$
- $\hat{y} \leq_{1\times 2} \hat{z} \iff \delta(M^Y(\alpha^z - \alpha^y) \mid \mathcal{B}^{n+1}) \leq |\beta^z| - |\beta^y|$

$\Longrightarrow$

$|\beta^z| \leq \max\{\delta(M^X(\alpha^z - \alpha^x) \mid \mathcal{B}^{n+1}) + |\beta^x|, \delta(M^Y(\alpha^z - \alpha^y) \mid \mathcal{B}^{n+1}) + |\beta^y|\}$

Which α^z minimizes $|\beta^z|$?

$\hat{z}$ is an upper bound :

- $\hat{x} \leq_{1\times 2} \hat{z} \iff \delta(M^X(\alpha^z - \alpha^x) \mid \mathcal{B}^{n+1}) \leq |\beta^z| - |\beta^x|$
- $\hat{y} \leq_{1\times 2} \hat{z} \iff \delta(M^Y(\alpha^z - \alpha^y) \mid \mathcal{B}^{n+1}) \leq |\beta^z| - |\beta^y|$

$\Longrightarrow$

$$|\beta^z| \leq \max\{\delta(M^X(\alpha^z - \alpha^x) \mid \mathcal{B}^{n+1}) + |\beta^x|, \delta(M^Y(\alpha^z - \alpha^y) \mid \mathcal{B}^{n+1}) + |\beta^y|\}$$

Which α^z minimizes $|\beta^z|$?

- $\hat{z}$ is an upper bound :
  - $|\beta^z| \leq \max\{\delta(M^X(\alpha^z - \alpha^x) \mid \mathcal{B}^{n+1}) + |\beta^x|, \delta(M^Y(\alpha^z - \alpha^y) \mid \mathcal{B}^{n+1}) + |\beta^y|\}$
- $\hat{z}$ is a minimal upper bound : we minimize this maximum

Characterization of α^z

- $|\beta^z|$ is a **saddle-value** of $L(\alpha, \lambda)$
- α^z is a **saddle-point** of $L(\alpha, \lambda)$

$L(\alpha, \lambda)$

$$L(\alpha, \lambda) = \lambda(\delta(M^X(\alpha - \alpha^x) \mid \mathcal{B}^{n+1}) + |\beta^x|) + (1-\lambda)(\delta(M^Y(\alpha - \alpha^y) \mid \mathcal{B}^{n+1}) + |\beta^y|),$$

where $\alpha \in \mathbb{R}^{n+1}$, and $\lambda \in [0, 1]$.

use the **subdifferential theory** and the **Fenchel duality**.

Which α^z minimizes $|\beta^z|$?

- $\hat{z}$ is an upper bound :
  - $|\beta^z| \leq \max\{\delta(M^X(\alpha^z - \alpha^x) \mid \mathcal{B}^{n+1}) + |\beta^x|, \delta(M^Y(\alpha^z - \alpha^y) \mid \mathcal{B}^{n+1}) + |\beta^y|\}$
- $\hat{z}$ is a minimal upper bound : we minimize this maximum

Characterization of α^z

- $|\beta^z|$ is a **saddle-value** of $L(\alpha, \lambda)$
- α^z is a **saddle-point** of $L(\alpha, \lambda)$

$L(\alpha, \lambda)$

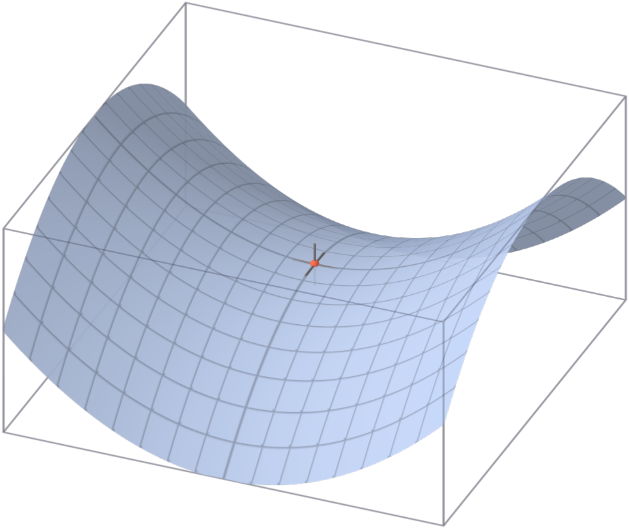
$$L(\alpha, \lambda) = \lambda(\delta(M^X(\alpha - \alpha^x) \mid \mathcal{B}^{n+1}) + |\beta^x|) \\ + (1 - \lambda)(\delta(M^Y(\alpha - \alpha^y) \mid \mathcal{B}^{n+1}) + |\beta^y|),$$

where $\alpha \in \mathbb{R}^{n+1}$, and $\lambda \in [0, 1]$.

use the **subdifferential theory** and the **Fenchel duality**.

Saddle-Point

$f(x, y) = x^2 - y^2$



Characterization of the set of saddle-points of $L(\alpha, \lambda)$

Theorem

When $\hat{x}$ and $\hat{y}$ are non comparable, we have

$$\bullet \bar{\lambda} \in]0, 1[,$$

$$\bullet \delta(M^X(\bar{\alpha} - \alpha^x) \mid \mathcal{B}^{n+1}) + |\beta^x| = \delta(M^Y(\bar{\alpha} - \alpha^y) \mid \mathcal{B}^{n+1}) + |\beta^y|,$$

$$\bullet \bar{\lambda}\delta(M^X(\bar{\alpha} - \alpha^x) \mid \mathcal{B}^{n+1}) + (1 - \bar{\lambda})\delta(M^Y(\bar{\alpha} - \alpha^y) \mid \mathcal{B}^{n+1}) = \delta(\alpha^x - \alpha^y \mid \bar{\lambda}M^{X^*}\mathcal{B}^{n+1} \cap (1 - \bar{\lambda})M^{Y^*}\mathcal{B}^{n+1}) \ .$$

where $(\bar{\alpha}, \bar{\lambda})$ denotes a saddle-point of L.

Complexity of Computations

$$\hat{X} = \begin{pmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_p \end{pmatrix}, \Phi^X \qquad \cup_{1\times 2} \qquad \hat{Y} = \begin{pmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_p \end{pmatrix}, \Phi^Y$$

$$=$$

$$\hat{Z} = \begin{pmatrix} \hat{z}_1 \\ \vdots \\ \hat{z}_p \end{pmatrix}, \Phi^X \cup_2 \Phi^Y$$

Complexity for each $\hat{z}_i$

- $\mathcal{O}(n^3)$ in the worst case to compute $|\beta^{z_i}|$,
- α^{z_i} is deduced as a solution of a LP of dimension $n+1$ with $2n+3$ constraints,

↠ **Polynomial** algorithm to compute a **minimal upper bound** with the **least perturbation**.

Other Join Variants

Two Sufficient Conditions for Minimality

① $|\beta^z|$ is minimal over all upper bounds ($\cup_{1\times 2}$)

② $|\beta^z|$ is minimal over all upper bounds which minimize the interval concretisation of $\hat{z}$ ($\sqcup_{1\times 2}$)

$\sqcup_{1\times 2}$

+ Computes a **minimal upper bound**
+ Linear Complexity
− May lose "many" noise symbols

$\uplus_{1\times 2}$ (weaker version of $\sqcup_{1\times 2}$)

+ Linear Complexity
+ lose less noise symbols
± Computes an upper bound in general (but may give the minimal upper bound returned by $\sqcup_{1\times 2}$)

Other Join Variants

Two Sufficient Conditions for Minimality

① $|\beta^z|$ is minimal over all upper bounds ($\cup_{1\times 2}$)

② $|\beta^z|$ is minimal over all upper bounds which minimize the interval concretisation of $\hat{z}$ ($\sqcup_{1\times 2}$)

$\sqcup_{1\times 2}$

+ Computes a **minimal upper bound**
+ Linear Complexity
− May lose "many" noise symbols

$\uplus_{1\times 2}$ (weaker version of $\sqcup_{1\times 2}$)

+ Linear Complexity
+ lose less noise symbols
± Computes an upper bound in general (but may give the minimal upper bound returned by $\sqcup_{1\times 2}$)

Other Join Variants

Two Sufficient Conditions for Minimality

① $|\beta^z|$ is minimal over all upper bounds ($\cup_{1\times 2}$)

② $|\beta^z|$ is minimal over all upper bounds which minimize the interval concretisation of $\hat{z}$ ($\sqcup_{1\times 2}$)

$\sqcup_{1\times 2}$

+ Computes a **minimal upper bound**
+ Linear Complexity
− May lose "many" noise symbols

$\uplus_{1\times 2}$ (weaker version of $\sqcup_{1\times 2}$)

+ Linear Complexity
+ lose less noise symbols
± Computes an upper bound in general (but may give the minimal upper bound returned by $\sqcup_{1\times 2}$)

Example 1

Minimal Upper Bounds

$$\begin{cases} \Phi^a = [-1, 0] \times [0, 0.5] \\ \hat{a} = 1 - \epsilon_1 + 2\epsilon_2 \\ \gamma(\hat{a}) = [1, 3] \end{cases} \quad \begin{cases} \Phi^b = [-0.5, 0.5] \times [0, 1] \\ \hat{b} = 2 + \epsilon_1 + \epsilon_2 \\ \gamma(\hat{b}) = [1.5, 3.5] \end{cases}$$

Two (non trivial) Minimal Upper Bounds:

$$\sqcup_{1\times 2}, \uplus_{1\times 2} \begin{cases} \Phi^c = [-1, 0.5] \times [0, 1] \\ \hat{c} = 1.75 + \epsilon_2 + 0.75\eta_u^c \\ \gamma(\hat{c}) = [1, 3.5] = [1, 3] \cup [1.5, 3.5] \end{cases}$$

$$\cup_{1\times 2} \begin{cases} \Phi^d = [-1, 0.5] \times [0, 1] \\ \hat{d} = 1.7 + 0.2\epsilon_1 + 1.6\epsilon_2 + 0.7\eta_u^d \\ \gamma(\hat{d}) = [0.8, 4.1] \supseteq [1, 3.5] \end{cases}$$

Example 2

Minimal Upper Bounds

$$\begin{cases} \Phi^a = [-1, 0] \times [0, 0.5] \\ \hat{a} = -2\epsilon_1 + \epsilon_2 \\ \gamma(\hat{a}) = [0, 2.5] \end{cases} \quad \begin{cases} \Phi^b = [-0.5, 0.5] \times [0, 1] \\ \hat{b} = -2\epsilon_1 + \epsilon_2 \\ \gamma(\hat{b}) = [-1, 2] \end{cases}$$

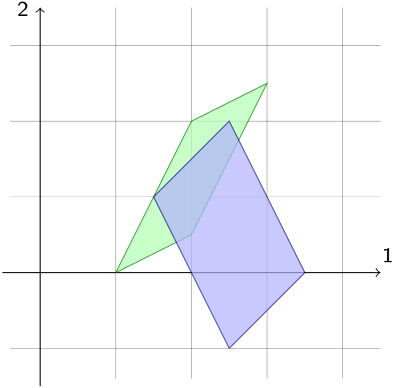
Two Minimal Upper Bounds:

$$\sqcup_{1\times 2} \begin{cases} \Phi^c = [-1, 0.5] \times [0, 1] \\ \hat{c} = 0.25 + \epsilon_2 + 1.25\eta_u^c \\ \gamma(\hat{c}) = [-1, 2.5] \end{cases} ; \cup_{1\times 2} \begin{cases} \Phi^d = [-1, 0.5] \times [0, 1] \\ \hat{d} = -2\epsilon_1 + \epsilon_2 \\ \gamma(\hat{d}) = [-1, 3] \end{cases}$$

An Upper Bound:

$$\uplus_{1\times 2} \begin{cases} \Phi^e = [-1, 0.5] \times [0, 1] \\ \hat{e} = 0.75 + 1.75\eta_u^c \\ \gamma(\hat{e}) = [-1, 2.5] = [0, 2.5] \cup [-1, 2] \end{cases}$$

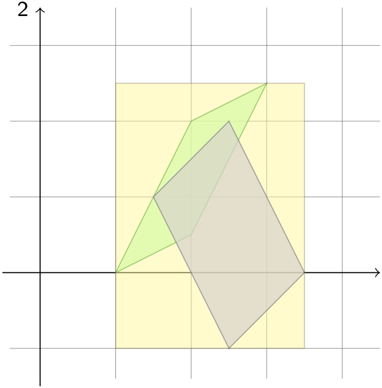
$\uplus_{1\times 2}$ vs $\sqcup_{1\times 2}$ vs $\cup_{1\times 2}$



$$\hat{X} = \begin{cases} \Phi^X = [-1, 0] \times [0, 0.5] \\ \hat{x}_1 = 1 - \epsilon_1 + 2\epsilon_2 \\ \hat{x}_2 = -2\epsilon_1 + \epsilon_2 \end{cases}$$

$$\hat{Y} = \begin{cases} \Phi^Y = [-0.5, 0.5] \times [0, 1] \\ \hat{y}_1 = 2 + \epsilon_1 + \epsilon_2 \\ \hat{y}_2 = -2\epsilon_1 + \epsilon_2 \end{cases}$$

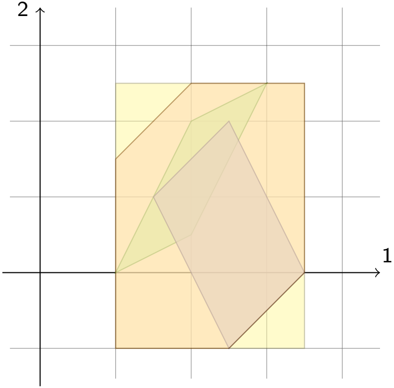
$\uplus_{1\times 2}$ vs $\sqcup_{1\times 2}$ vs $\cup_{1\times 2}$



$$\Phi^Z = [-0.5, 1] \times [0, 1]$$

$$\uplus_{1\times 2} \begin{cases} \hat{z}_1 = 1.75 + \epsilon_2 + 0.75\eta_u^{z_1} \\ \hat{z}_2 = 0.75 + 1.75\eta_u^{z_2} \end{cases}$$

$\uplus_{1\times 2}$ vs $\sqcup_{1\times 2}$ vs $\cup_{1\times 2}$



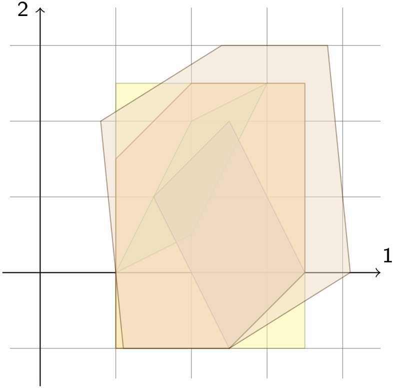
$$\Phi^Z = [-0.5, 1] \times [0, 1]$$

$$\uplus_{1\times 2} \begin{cases} \hat{z}_1 = 1.75 + \epsilon_2 + 0.75\eta_u^{z_1} \\ \hat{z}_2 = 0.75 + 1.75\eta_u^{z_2} \end{cases}$$

$$\sqcup_{1\times 2} \begin{cases} \hat{z}_1 = 1.75 + \epsilon_2 + 0.75\eta_u^{z_1} \\ \hat{z}_2 = 0.25 + \epsilon_2 + 1.25\eta_u^{z_2} \end{cases}$$

$\uplus_{1\times 2}$ vs $\sqcup_{1\times 2}$ vs $\cup_{1\times 2}$

$$\Phi^Z = [-0.5, 1] \times [0, 1]$$

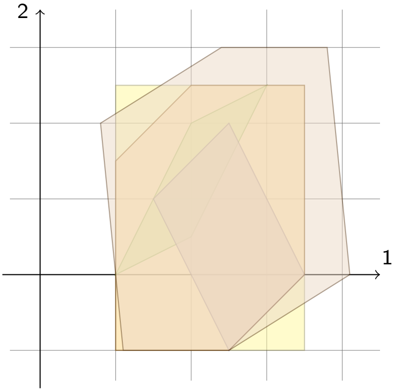


$$\uplus_{1\times 2} \begin{cases} \hat{z}_1 = 1.75 + \epsilon_2 + 0.75\eta_u^{z_1} \\ \hat{z}_2 = 0.75 + 1.75\eta_u^{z_2} \end{cases}$$

$$\sqcup_{1\times 2} \begin{cases} \hat{z}_1 = 1.75 + \epsilon_2 + 0.75\eta_u^{z_1} \\ \hat{z}_2 = 0.25 + \epsilon_2 + 1.25\eta_u^{z_2} \end{cases}$$

$$\cup_{1\times 2} \begin{cases} \hat{z}_1 = 1.7 + 0.2\epsilon_1 + 1.6\epsilon_2 + 0.7\eta_u^{z_1} \\ \hat{z}_2 = -2\epsilon_1 + \epsilon_2 \end{cases}$$

$\uplus_{1\times 2}$ vs $\sqcup_{1\times 2}$ vs $\cup_{1\times 2}$



$$\Phi^Z = [-0.5, 1] \times [0, 1]$$

$$\uplus_{1\times 2} \begin{cases} \hat{z}_1 = 1.75 + \epsilon_2 + 0.75\eta_u^{z_1} \\ \hat{z}_2 = 0.75 + 1.75\eta_u^{z_2} \end{cases}$$

$$\sqcup_{1\times 2} \begin{cases} \hat{z}_1 = 1.75 + \epsilon_2 + 0.75\eta_u^{z_1} \\ \hat{z}_2 = 0.25 + \epsilon_2 + 1.25\eta_u^{z_2} \end{cases}$$

- We use in addition a Reduced Product with Intervals.

$$\cup_{1\times 2} \begin{cases} \hat{z}_1 = 1.7 + 0.2\epsilon_1 + 1.6\epsilon_2 + 0.7\eta_u^{z_1} \\ \hat{z}_2 = -2\epsilon_1 + \epsilon_2 \end{cases}$$

Interpretation of Tests

$$\hat{X} = \begin{pmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_p \end{pmatrix}, \Phi^X \qquad \{\hat{x}_i == \hat{x}_j,\ \hat{x}_i \leq 0,\ \hat{x}_j \times \hat{x}_i \leq \hat{x}_k\} \qquad \hat{Y} = \begin{pmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_p \end{pmatrix}, \Phi^Y$$

$$\overset{\Delta}{\Longleftrightarrow}$$

$$\Phi^Y = [\![cons]\!]_2^\sharp(\Phi^X)$$

$$\forall i, 1 \leq i \leq p, \quad \hat{y}_i = \hat{x}_i$$

Interpretation of Equality Tests

Example

$$\hat{X} = \begin{pmatrix} \hat{x}_1 = 4 + \epsilon_1 + \epsilon_2 + \epsilon_3 \\ \hat{x}_2 = \quad - \epsilon_1 + 3\epsilon_2 \\ \hat{x}_3 = \quad - \epsilon_1 + 2\epsilon_2 + \epsilon_3 \end{pmatrix}, \Phi^X = [-1, 1]^3 \qquad \{\hat{x}_1 == \hat{x}_2\} \qquad \hat{Y} = ?$$

- $\hat{x}_1 == \hat{x}_2 \iff 4 + 2\epsilon_1 - 2\epsilon_2 + \epsilon_3 == 0,$
- $\Phi^Y = [\![4 + 2\epsilon_1 - 2\epsilon_2 + \epsilon_3 == 0]\!]^\sharp_2(\Phi^X) = [-1, -0.5] \times [0.5, 1] \times [-1, 0]$
- $\hat{y}_i$ is extracted from $\hat{x}_i$ such that the interval concretisation of $\hat{y}_i$ is minimal

Interpretation of Equality Tests

Example

$$\hat{X} = \begin{pmatrix} \hat{x}_1 = 4 + \epsilon_1 + \epsilon_2 + \epsilon_3 \\ \hat{x}_2 = \quad - \epsilon_1 + 3\epsilon_2 \\ \hat{x}_3 = \quad - \epsilon_1 + 2\epsilon_2 + \epsilon_3 \end{pmatrix}, \Phi^X = [-1, 1]^3 \qquad \{\hat{x}_1 == \hat{x}_2\} \qquad \hat{Y} = ?$$

- $\hat{x}_1 == \hat{x}_2 \iff 4 + 2\epsilon_1 - 2\epsilon_2 + \epsilon_3 == 0$,
- $\Phi^Y = [\![4 + 2\epsilon_1 - 2\epsilon_2 + \epsilon_3 == 0]\!]_2^\sharp(\Phi^X) = [-1, -0.5] \times [0.5, 1] \times [-1, 0]$
- $\hat{y}_i$ is extracted from $\hat{x}_i$ such that the interval concretisation of $\hat{y}_i$ is minimal

Interpretation of Equality Tests

Example

$$\hat{X} = \begin{pmatrix} \hat{x}_1 = 4 + \epsilon_1 + \epsilon_2 + \epsilon_3 \\ \hat{x}_2 = \quad - \epsilon_1 + 3\epsilon_2 \\ \hat{x}_3 = \quad - \epsilon_1 + 2\epsilon_2 + \epsilon_3 \end{pmatrix}, \Phi^X = [-1, 1]^3 \qquad \{\hat{x}_1 == \hat{x}_2\} \qquad \hat{Y} = ?$$

- $\hat{x}_1 == \hat{x}_2 \iff 4 + 2\epsilon_1 - 2\epsilon_2 + \epsilon_3 == 0,$
- $\Phi^Y = [\![4 + 2\epsilon_1 - 2\epsilon_2 + \epsilon_3 == 0]\!]^\sharp_2(\Phi^X) = [-1, -0.5] \times [0.5, 1] \times [-1, 0]$
- $\hat{y}_i$ is extracted from $\hat{x}_i$ such that the interval concretisation of $\hat{y}_i$ is minimal

Interpretation of Equality Tests

Example Con't

$$\hat{y}_1 = 2 + 2\epsilon_2 + 0.5\epsilon_3, \quad \text{bound}_2(\hat{Y}_1, \Phi^Y) = [2.5, 4] \quad \text{(by substituting } \epsilon_1)$$

$$\hat{y}_1 = 6 + 2\epsilon_1 + 1.5\epsilon_1, \quad \text{bound}_2(\hat{Y}_1, \Phi^Y) = [2.5, 5] \quad \text{(by substituting } \epsilon_2)$$

$$\hat{y}_1 = -\epsilon_1 + 3\epsilon_2, \quad \text{bound}_2(\hat{Y}_1, \Phi^Y) = [2, 4] \quad \text{(by substituting } \epsilon_3)$$

$$\Phi^Y := [-1, -0.5] \times [0.5, 1] \times [-1, 0]$$

$$\hat{y}_1 := 2 + 2\epsilon_2 + 0.5\epsilon_3, \quad \text{bound}_2(\hat{Y}_1, \Phi^Y) = [2.5, 4]$$

$$\hat{y}_2 := 2 + 2\epsilon_2 + 0.5\epsilon_3, \quad \text{bound}_2(\hat{Y}_2, \Phi^Y) = [2.5, 4]$$

$$\hat{y}_3 := 2 + \epsilon_2 + 1.5\epsilon_3, \quad \text{bound}_2(\hat{Y}_3, \Phi^Y) = [1, 3]$$

Interpretation of Equality Tests

Example Con't

$$\hat{y}_1 = 2 + 2\epsilon_2 + 0.5\epsilon_3, \quad \mathrm{bound}_2(\hat{Y}_1, \Phi^Y) = [2.5, 4] \quad \text{(by substituting } \epsilon_1\text{)}$$

$$\hat{y}_1 = 6 + 2\epsilon_1 + 1.5\epsilon_1, \quad \mathrm{bound}_2(\hat{Y}_1, \Phi^Y) = [2.5, 5] \quad \text{(by substituting } \epsilon_2\text{)}$$

$$\hat{y}_1 = -\epsilon_1 + 3\epsilon_2, \quad \mathrm{bound}_2(\hat{Y}_1, \Phi^Y) = [2, 4] \quad \text{(by substituting } \epsilon_3\text{)}$$

$$\Phi^Y := [-1, -0.5] \times [0.5, 1] \times [-1, 0]$$

$$\hat{y}_1 := 2 + 2\epsilon_2 + 0.5\epsilon_3, \quad \mathrm{bound}_2(\hat{Y}_1, \Phi^Y) = [2.5, 4]$$

$$\hat{y}_2 := 2 + 2\epsilon_2 + 0.5\epsilon_3, \quad \mathrm{bound}_2(\hat{Y}_2, \Phi^Y) = [2.5, 4]$$

$$\hat{y}_3 := 2 + \epsilon_2 + 1.5\epsilon_3, \quad \mathrm{bound}_2(\hat{Y}_3, \Phi^Y) = [1, 3]$$

Interpretation of Equality Tests

Example Con't

Properties

- for each i, we solve the above problem with an average complexity of $\mathcal{O}(n \log(n))$,
- the equality constraint is **algebraically satisfied** in $\hat{Y}$: $\hat{y}_1 = \hat{y}_2$,
- the concretisation of each $\hat{y}_i$ is optimal.

Assignment - Widening

Assignment

- Φ is unchanged
- For linear expressions, we simply use affine arithmetic as in Perturbed Affine Sets (Φ is unused)
- For non linear operations, Φ **is used** to improve the linearization of non linear terms

Widening

We use the same widening as in Perturbed Affine Sets: losing the relations encoded by noise symbols.

Outlines

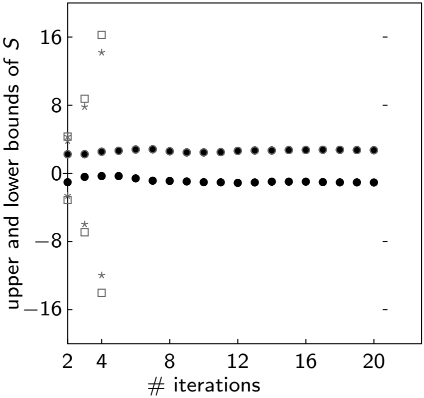
Taylor1+ Features

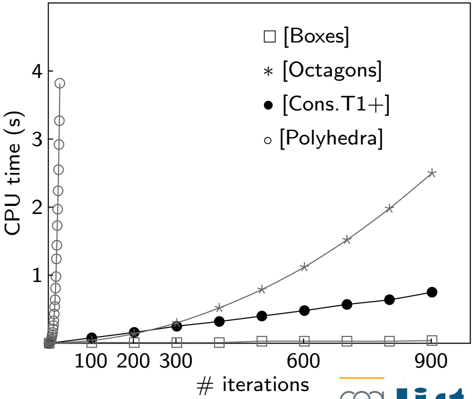
- analyses programs with **real number semantics**,
- APRON [B. Jeannet, A.Miné, SAS07] like abstract domain (level 0),
- written in C and offers an OCAML interface,
- linked to interproc [B. Jeannet],
- uses **double-precision floating-point numbers** in a **sound** manner for computations in abstract domain (supports also GMP and MPFR),
- Noise symbols abstract domains may be any APRON like abstract domain.

Unrolled scheme for the 2^{nd} order filter

2^{nd} order filter

- $S_n = 0.7E_n - 1.3E_{n-1} + 1.1E_{n-2} + 1.4S_{n-1} - 0.7S_{n-2}$
- Poles are inside the unit circle (norm close to 0.84)





Fixpoint Computation

filter o2	fixpoint	t(s)
Boxes	$\top$	6×10^{-3}
Octagons	$\top$	0.19
Polyhedra	$[-1.30\ ,\ 2.82]$	0.49
Taylor1+	$[-5.40\ ,\ 7.07]$	0.2

filter o8	fixpoint	t(s)
Boxes	$\top$	0.01
Octagons	$\top$	21
Polyhedra	abort	$>24h$
Taylor1+	$[-3.81\ ,\ 4.81]$	0.5

3^{rd} order Householder Iteration Scheme

Inverse of the square root

- $h_n = 1 - Ax_n^2$, $A \in [16, 20]$ and $x_0 = 2^{-4}$

$$x_{n+1} = x_n + x_n \left(\frac{1}{2} h_n + \frac{3}{8} h_n^2 \right)$$

Unrolling (5 It.)	$\sqrt{A} = Ax_n$	t(s)
Boxes	[0.51 , 8.44]	1×10^{-4}
Octagons	[0.51 , 7.91]	0.01
Polyhedra	[2.22 , 6.56]	310
T.1+ :	[3.97 , 4.51]	1×10^{-3}
• 10 subdivisions	[4.00 , 4.47]	0.02
• SDP	[3.97 , 4.51]	0.16

Benchmarks

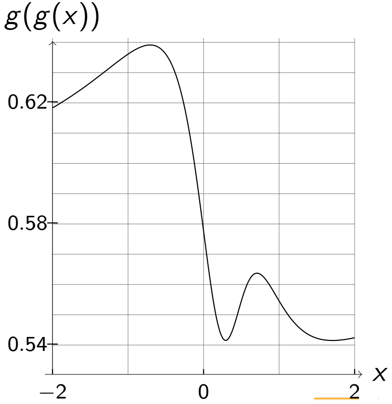
- `InterQ1`: linear tests with quadratic expressions
- `Cosine`: piecewise 3rd order polynomial interpolation of the cosine function
- `SinCos`: sum of the squares of the sine and cosine functions
- `InterL2` (resp. `InterQ2`): the inverse image of 1 by a piecewise affine (resp. quadratic) function

	Exact	Octagons	Polyhedra	Taylor1+	**Cons. Taylor1+** ($\uplus_{1\times 2}$)
InterQ1	$[\mathbf{0}, \mathbf{1875}]$	$[-3750, 6093]$	$[-2578, 4687]$	$[0, 2500]$	$[0, 1875]$
Cosine	$[-\mathbf{1}, \mathbf{1}]$	$[-1.50, 1.0]$	$[-1.50, 1.0]$	$[-1.073, 1]$	$[-1, 1]$
SinCos	$\{\mathbf{1}\}$	$[0.84, 1.15]$	$[0.91, 1.07]$	$[0.86, 1.15]$	$[0.99, 1.00]$
InterL2	$\{\mathbf{0.1}\}$	$[-1, 1]$	$[0.1, 0.4]$	$[-1, 1]$	$[0.1, 1]$
InterQ2	$\{\mathbf{0.36}\}$	$[-1, 1]$	$[-0.8, 1]$	$[-1, 1]$	$[-0.4, 1]$
InterQ2b	$[-\mathbf{0.1}, \mathbf{3}]$	$[-3, 27]$	$[-3, 27]$	$[-0.1, 27]$	$[-0.1, 3.77]$

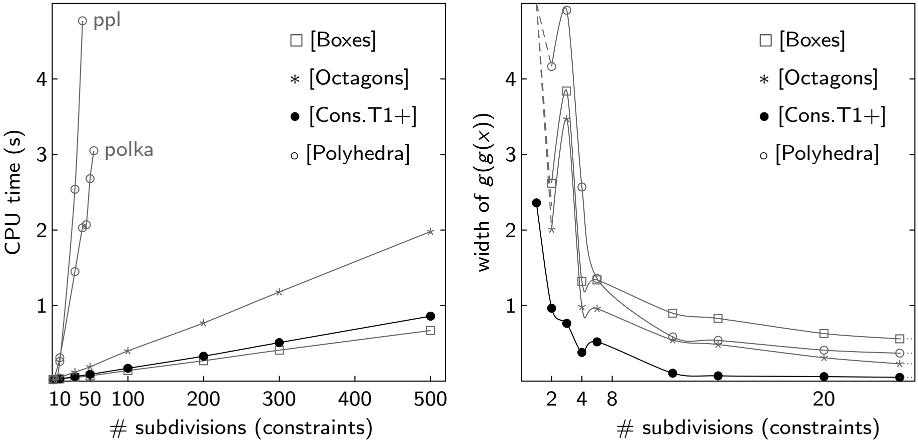
Does the domain scale up ?

$$g(x) = \frac{\sqrt{x^2 - x + 0.5}}{\sqrt{x^2 + 0.5}}$$

```
g(x) = sqrt(x*x-x+0.5)/sqrt(x*x+0.5);
x = [-2,2];
/* for n subdivisions */
h = 4/n;
if (-x<=h-2)
  y = g(x); z = g(y);
...
/* 2 <= i <= n-1 */
else if (-x<=i*h-2)
  y = g(x); z = g(y);
...
else
  y = g(x); z = g(y);
```



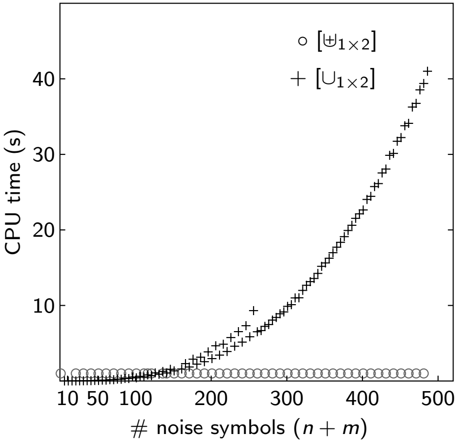
Taylor1+ Scales up



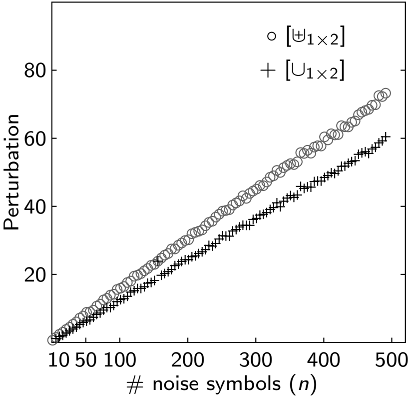
Comparison of Join Variants

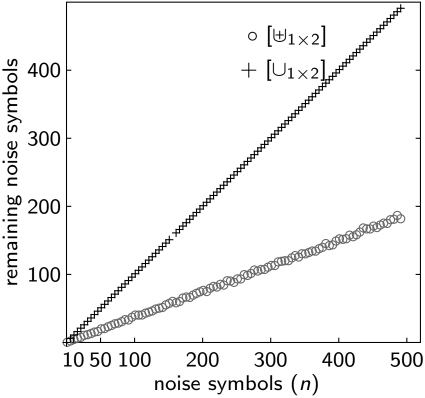
	Exact	Taylor1+	Cons. Taylor1+ ($\uplus_{1\times2}$)	Cons. Taylor1+ ($\cup_{1\times2}$)
InterQ1	$[\mathbf{0}, \mathbf{1875}]$	$[0, 2500]$	$[0, 1875]$	$[0, 1875]$
Cosine	$[-\mathbf{1}, \mathbf{1}]$	$[-1.073, 1]$	$[-1, 1]$	$[-1, 1]$
SinCos	$\{\mathbf{1}\}$	$[0.86, 1.15]$	$[0.99, 1.00]$	$[0.99, 1.00]$
InterL2	$\{\mathbf{0.1}\}$	$[-1, 1]$	$[0.1, 1]$	$[0.066, 0.4]$
InterQ2	$\{\mathbf{0.36}\}$	$[-1, 1]$	$[-0.4, 1]$	$[-0.29, 0.52]$
InterQ2b	$[-\mathbf{0.1}, \mathbf{3}]$	$[-0.1, 27]$	$[-0.1, 3.77]$	$[-0.1, 3.77]$

Scalabity of Join Operators



Perturbation and Lost Noise Symbols





Conclusion

Future Directions

- Using **non linear templates** abstract domain to abstract noise symbols (back to the quaternion normalization problem, we can use the abstract domain [Adjé,Gaubert,Goubault] for noise symbols).
- **Abstract the coefficients** to catch some specific non-convex (disjunctive) properties.
- better global join than the diagonal relaxiation (we have already promising results).

Thanks for your attention !

Kolev Multiplication

- no overestimation if certain simple monotonicity conditions are valid [Kolev 2007].
- However, the affine form obtained is not always correct when dealing with future evaluations.

Example

- $\hat{x} = 10 + 5\epsilon_1 + 3\epsilon_2$ and $\hat{y} = 10 - 2\epsilon_1 + \epsilon_3$,
- Kolev multiplication gives $\hat{z} = 92 + 31\epsilon_1 + 21\epsilon_2 + 2\epsilon_3 + 16\epsilon_4$.
- $\gamma(z)$ is $[22, 162]$ which is the exact range of xy.
- for $t = -4x + 0.8z - 79$,
- $\hat{t} = -45.4 + 4.8\epsilon_1 + 4.8\epsilon_2 + 1.6\epsilon_3 + 12.8\epsilon_4$,
- and $\gamma(t) \in [-69.4, -21.4]$.
- for $\epsilon_1 = 0$ and $\epsilon_2 = 1$ and $\epsilon_3 = 1$,
- $x = 13$ and $y = 11$ and $z = 143$, then $t = -16.6 \notin \gamma(t)$.

affine forms multiplication

Lemma [Gaubert 2006]

$$\max_{|\epsilon_i|\leq 1} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i^x \alpha_j^y \epsilon_i \epsilon_j = \max_{|\epsilon_i|\leq 1} \varepsilon^t.\Phi.\varepsilon \leq \inf_{\mu \in \mathbb{R}_+^n} \{\mathrm{trace}(\mu I_n) | \Phi - \mu I_n \preceq 0\} \quad (1)$$

where

- $(\phi_{i,j})_{1\leq i,j\leq n} = \frac{1}{2}(\alpha_i^x \alpha_j^y + \alpha_j^x \alpha_i^y)$
- $M \preceq 0$ (M is negative semidefinite)

The equality holds when matrix Φ is negative semidefinite. The right hand side of (1) is a typical SDP problem.

using SDP throw an example

Let $\hat{x} = 10 + 5\epsilon_1 + 3\epsilon_2$ and $\hat{y} = 10 - 2\epsilon_1 + \epsilon_3$, then
$\hat{z} = \hat{x} \times \hat{y} = 100 + 30\epsilon_1 + 30\epsilon_2 + 10\epsilon_3 + q(\epsilon)$, where $q(\epsilon) = \epsilon^t Q \epsilon$ and

$$Q = \begin{pmatrix} -10 & -3 & 2.5 \\ -3 & 0 & 1.5 \\ 2.5 & 1.5 & 0 \end{pmatrix}$$

SDP problems to solve are:

$$\mathbf{1)} \quad \begin{array}{ll} M & = \min \mu.1_n \\ s.t & \begin{pmatrix} \mu I_n - Q & 0 \\ 0 & \mu I_n \end{pmatrix} \succeq 0 \end{array} \qquad \mathbf{2)} \quad \begin{array}{ll} -m & = \min \mu.1_n \\ s.t & \begin{pmatrix} \mu I_n - (-Q) & 0 \\ 0 & \mu I_n \end{pmatrix} \succeq 0 \end{array}$$

The final invariant of InterQ2