

MPRI

# The Arithmetic-Geometric Progression Abstract Domain

VMCAI 2005

Jérôme Feret

Laboratoire d'Informatique de l'École Normale Supérieure  
INRIA, ÉNS, CNRS

<http://www.di.ens.fr/~feret>

November, 2023

# Overview

1. Introduction
2. Case study
3. Arithmetic-geometric progressions
4. Benchmarks
5. Conclusion

# Issue

- In automatically generated programs using floating point arithmetics, some computations may diverge because of rounding errors.
- We prove the absence of floating point number overflows: we bound rounding errors at each loop iteration by a linear combination of the loop inputs; we get bounds on the values that depends exponentially on the program execution time.
- We use non polynomial constraints. Our domain is both precise (no false alarm) and efficient (linear in memory /  $n \ln(n)$  in time).

# Overview

1. Introduction
2. Case study
3. Arithmetic-geometric progressions
4. Benchmarks
5. Conclusion

# Running example (in $\mathbb{R}$ )

1 :  $X := 0; k := 0;$

2 : **while** ( $k < 1000$ ) {

3 :   **if** (?) { $X \in [-10; 10]$ };

4 :    $X := X/3;$

5 :    $X := 3 \times X;$

6 :    $k := k + 1;$

7 : }

# Interval analysis: first loop iteration

1 :  $X := 0; k := 0;$

$$X = 0$$

2 : **while** ( $k < 1000$ ) {

$$X = 0$$

3 :   **if** (?) { $X \in [-10; 10]$ };

$$|X| \leq 10$$

4 :    $X := X/3;$

$$|X| \leq \frac{10}{3}$$

5 :    $X := 3 \times X;$

$$|X| \leq 10$$

6 :    $k := k + 1;$

7 : }

# Interval analysis: Invariant

1 :  $X := 0; k := 0;$

$$X = 0$$

2 : **while** ( $k < 1000$ ) {

$$|X| \leq 10$$

3 :   **if** (?) { $X \in [-10; 10]$ };

$$|X| \leq 10$$

4 :    $X := X/3;$

$$|X| \leq \frac{10}{3}$$

5 :    $X := 3 \times X;$

$$|X| \leq 10$$

6 :    $k := k + 1;$

7 : }

$$|X| \leq 10$$

# Including rounding errors [Miné–ESOP'04]

```
1 : X := 0; k := 0;
2 : while (k < 1000) {
3 :   if (?) {X ∈ [-10; 10]};
4 :   X := X/3 + [-ε1; ε1].X + [-ε2; ε2];
5 :   X := 3 × X + [-ε3; ε3].X + [-ε4; ε4];
6 :   k := k + 1;
7 : }
```

The constants  $\varepsilon_1$ ,  $\varepsilon_2$ ,  $\varepsilon_3$ , and  $\varepsilon_4$  ( $\geq 0$ ) are computed by other domains.



# Interval analysis

Let  $M \geq 0$  be a bound:

1 :  $X := 0; k := 0;$

$$X = 0$$

2 : **while** ( $k < 1000$ ) {

$$|X| \leq M$$

3 :   **if** (?) { $X \in [-10; 10]$ };

$$|X| \leq \max(M, 10)$$

4 :    $X := X/3 + [-\varepsilon_1; \varepsilon_1].X + [-\varepsilon_2; \varepsilon_2];$

$$|X| \leq (\varepsilon_1 + \frac{1}{3}) \times \max(M, 10) + \varepsilon_2$$

5 :    $X := 3 \times X + [-\varepsilon_3; \varepsilon_3].X + [-\varepsilon_4; \varepsilon_4];$

$$|X| \leq (1 + \alpha) \times \max(M, 10) + b$$

6 :    $k := k + 1;$

7 :   }

with  $\alpha = 3 \times \varepsilon_1 + \frac{\varepsilon_3}{3} + \varepsilon_1 \times \varepsilon_3$  and  $b = \varepsilon_2 \times (3 + \varepsilon_3) + \varepsilon_4$ .

# Ari.-geo. analysis: first iteration

1 :  $X := 0; k := 0;$

$$X = 0, k = 0$$

2 : **while** ( $k < 1000$ ) {

$$X = 0$$

3 :   **if** (?) { $X \in [-10; 10]$ };

$$|X| \leq 10$$

4 :    $X := X/3 + [-\varepsilon_1; \varepsilon_1].X + [-\varepsilon_2; \varepsilon_2];$

$$|X| \leq \left[ v \mapsto \left( \frac{1}{3} + \varepsilon_1 \right) \times v + \varepsilon_2 \right] (10)$$

5 :    $X := 3 \times X + [-\varepsilon_3; \varepsilon_3].X + [-\varepsilon_4; \varepsilon_4];$

$$|X| \leq f^{(1)}(10)$$

6 :    $k := k + 1;$

$$|X| \leq f^{(k)}(10), k = 1$$

7 :   }

with  $f = \left[ v \mapsto \left( 1 + 3 \times \varepsilon_1 + \frac{\varepsilon_3}{3} + \varepsilon_1 \times \varepsilon_3 \right) \times v + \varepsilon_2 \times (3 + \varepsilon_3) + \varepsilon_4 \right].$

# Ari.-geo. analysis: Invariant

1 :  $X := 0; k := 0;$

$$X = 0, k = 0$$

2 : **while** ( $k < 1000$ ) {

$$|X| \leq f^{(k)}(10)$$

3 :   **if** (?) { $X \in [-10; 10]$ };

$$|X| \leq f^{(k)}(10)$$

4 :    $X := X/3 + [-\varepsilon_1; \varepsilon_1].X + [-\varepsilon_2; \varepsilon_2];$

$$|X| \leq \left(\frac{1}{3} + \varepsilon_1\right) \times \left(f^{(k)}(10)\right) + \varepsilon_2$$

5 :    $X := 3 \times X + [-\varepsilon_3; \varepsilon_3].X + [-\varepsilon_4; \varepsilon_4];$

$$|X| \leq f\left(f^{(k)}(10)\right)$$

6 :    $k := k + 1;$

$$|X| \leq f^{(k)}(10)$$

7 :   }

$$|X| \leq f^{(1000)}(10)$$

with  $f = \left[ v \mapsto \left(1 + 3 \times \varepsilon_1 + \frac{\varepsilon_3}{3} + \varepsilon_1 \times \varepsilon_3\right) \times v + \varepsilon_2 \times (3 + \varepsilon_3) + \varepsilon_4 \right].$

# Analysis session

The screenshot shows the Visualizator application window. The title bar reads "Visualizator". The menu bar contains icons for Quit, Intervals, Clocks, Trees, Octagons, Filters, Geom. dev., Symbolics, and Help. Below the menu bar is a search string field and navigation buttons: Next, Previous, First, Last, and Goto line: (with a dropdown). Below that is a "Program points:" section with buttons for Current, Next, Prev, Step, Backstep, and Variables: (with a dropdown set to "All" and a "Choose..." button).

The main editor area shows the following C code in a file named "example2.c":

```
void main()
{
  a = -10; b = 10; alpha = 3;
  while ((1 == 1))
  {
    if (B1) {X=NUM_input;};
    X = X/alpha;
    X = X*alpha;
    __ASTREE_wait_for_clock ();
  }
}
```

Below the code editor, the status bar displays the following information:

location: example2.c:14:33:[call#main@8:loop@10>=4:]  
variables: X (10)  
invariant:  
 $|X| \leq (10. + 2.35098891184e-38/(1.00000023842-1))*(1.00000023842)^{\text{clock}} - 2.35098891184e-38/(1.00000023842-1)$   
 $\leq 23.5916342108$

At the bottom of the status bar, it shows "example2.c — line 14 — column 33 — character 316".

# Overview

1. Introduction
2. Case study
3. Arithmetic-geometric progressions
4. Benchmarks
5. Conclusion

# Arithmetic-geometric progressions (in $\mathbb{R}$ )

An **arithmetic-geometric progression** is a 5-tuple in  $(\mathbb{R}^+)^5$ .

An arithmetic-geometric progression denotes a function in  $\mathbb{N} \rightarrow \mathbb{R}^+$ :

$$\beta_{\mathbb{R}}(M, a, b, a', b')(k) \triangleq [v \mapsto a \times v + b] \left( [v \mapsto a' \times v + b']^{(k)}(M) \right)$$

Thus,

- $k$  is the loop counter;
- $M$  is an initial value;
- $[v \mapsto a \times v + b]$  describes the current iteration;
- $[v \mapsto a' \times v + b']^{(k)}$  describes the first  $k$  iterations.

A **concretization**  $\gamma_{\mathbb{R}}$  maps each element  $d \in (\mathbb{R}^+)^5$  to a set  $\gamma_{\mathbb{R}}(d) \subseteq (\mathbb{N} \rightarrow \mathbb{R}^+)$  defined as:

$$\{f \mid \forall k \in \mathbb{N}, |f(k)| \leq \beta_{\mathbb{R}}(d)(k)\}$$

# Monotonicity

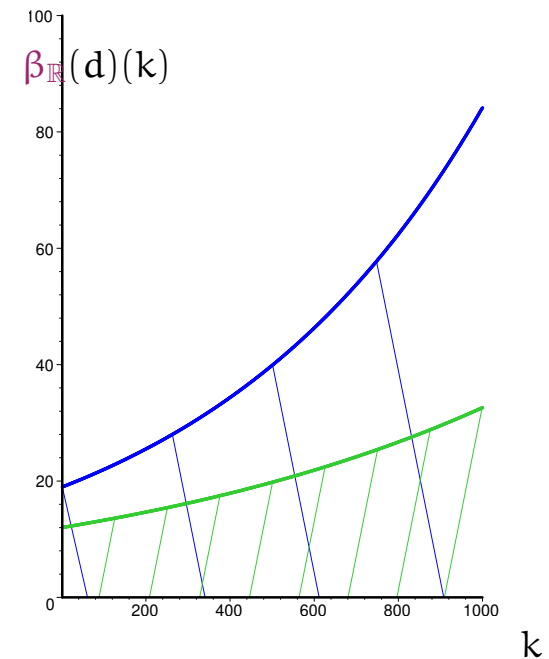
Let  $d = (M, a, b, a', b')$  and  $d = (M, a, b, a', b')$  be two arithmetic-geometric progressions.

If:

- $M \leq M$ ,
- $a \leq a, a' \leq a'$ ,
- $b \leq b, b' \leq b'$ .

Then:

$$\forall k \in \mathbb{N}, \beta_{\mathbb{R}}(d)(k) \leq \beta_{\mathbb{R}}(d)(k).$$



# Disjunction

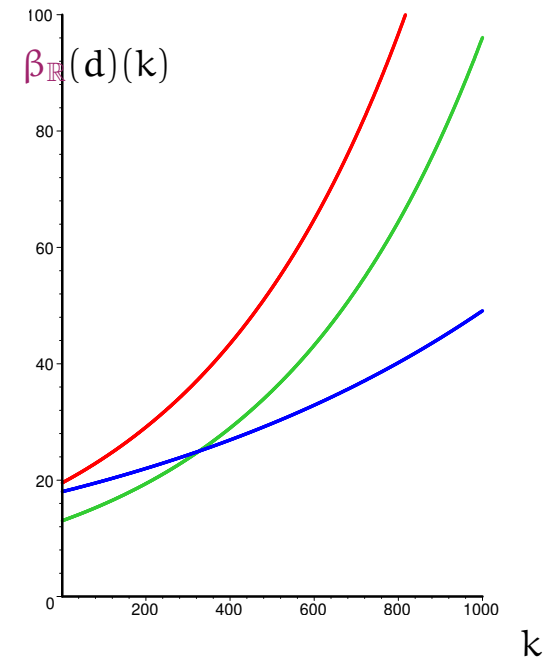
Let  $d = (M, a, b, a', b')$  and  $d = (M, a, b, a', b')$  be two arithmetic-geometric progressions.

We define:

$$d \sqcup_{\mathbb{R}} d \triangleq (M, a, b, a', b')$$

where:

- $M \triangleq \max(M, M)$ ,
- $a \triangleq \max(a, a)$ ,  $a' \triangleq \max(a', a')$ ,
- $b \triangleq \max(b, b)$ ,  $b' \triangleq \max(b', b')$ ,



For any  $k \in \mathbb{N}$ ,  $\beta_{\mathbb{R}}(d \sqcup_{\mathbb{R}} d)(k) \geq \max(\beta_{\mathbb{R}}(d)(k), \beta_{\mathbb{R}}(d)(k))$ .



# Conjunction

Let  $\mathbf{d}$  and  $\mathbf{d}$  be two arithmetic-geometric progressions.

1. If  $\mathbf{d}$  and  $\mathbf{d}$  are comparable (component-wise), we take the smaller one:

$$\mathbf{d} \sqcap_{\mathbb{R}} \mathbf{d} \stackrel{\Delta}{=} \text{Inf}_{\leq} \{\mathbf{d}; \mathbf{d}\}.$$

2. Otherwise, we use a parametric strategy:

$$\mathbf{d} \sqcap_{\mathbb{R}} \mathbf{d} \in \{\mathbf{d}; \mathbf{d}\}.$$

For any  $k \in \mathbb{N}$ ,  $\beta_{\mathbb{R}}(\mathbf{d} \sqcap_{\mathbb{R}} \mathbf{d})(k) \geq \min(\beta_{\mathbb{R}}(\mathbf{d})(k), \beta_{\mathbb{R}}(\mathbf{d})(k))$ .

# Assignment (I/III)

We have:

$$\begin{aligned}\beta_{\mathbb{R}}(M, a, b, a', b')(k) &= a \times (M + b' \times k) + b && \text{when } a' = 1 \\ \beta_{\mathbb{R}}(M, a, b, a', b')(k) &= a \times \left( (a')^k \times \left( M - \frac{b'}{1-a'} \right) + \frac{b'}{1-a'} \right) + b && \text{when } a' \neq 1.\end{aligned}$$

Thus:

1. for any  $a, a', M, b, b', \lambda \in \mathbb{R}^+$ ,

$$\lambda \times \left( \beta_{\mathbb{R}}(M, a, b, a', b')(k) \right) = \beta_{\mathbb{R}}(\lambda \times M, a, \lambda \times b, a', \lambda \times b')(k);$$

2. for any  $a, a', M, b, b', M, b, b' \in \mathbb{R}^+$ , for any  $k \in \mathbb{N}$ ,

$$\beta_{\mathbb{R}}(M, a, b, a', b')(k) + \beta_{\mathbb{R}}(M, a, b, a', b')(k) = \beta_{\mathbb{R}}(M + M, a, b + b, a', b' + b')(k).$$

# Assignment (II/III)

For  $k \in \mathbb{N}$ , if:

$$|X_i| \leq \beta_{\mathbb{R}}(M_i, \alpha_i, b_i, \alpha'_i, b'_i) (k)$$

then:

$$\frac{|B + \sum \alpha_i \times X_i| - |B|}{\sum |\alpha_i|} \leq \beta_{\mathbb{R}}\left(\frac{\sum |\alpha_i| \times M_i}{\sum |\alpha_i|}, \text{Max}(\alpha_i), \frac{\sum |\alpha_i| \times b_i}{\sum |\alpha_i|}, \text{Max}(\alpha'_i), \frac{\sum |\alpha_i| \times b'_i}{\sum |\alpha_i|}\right) (k)$$

so:

$$|B + \sum \alpha_i \times X_i| \leq \beta_{\mathbb{R}}\left(\frac{\sum |\alpha_i| \times M_i}{\sum |\alpha_i|}, \sum |\alpha_i| \times \text{Max}(\alpha_i), \frac{\sum |\alpha_i| \times b_i}{\sum |\alpha_i|} + |B|, \text{Max}(\alpha'_i), \frac{\sum |\alpha_i| \times b'_i}{\sum |\alpha_i|}\right) (k)$$

# Assignment (III/III)

If for  $k \in \mathbb{N}$ ,  $|X| \leq \beta_{\mathbb{R}}(M_X, a_X, b_X, a'_X, b'_X)(k)$  and  $|Y| \leq \beta_{\mathbb{R}}(M_Y, a_Y, b_Y, a'_Y, b'_Y)(k)$ , then:

1. increment:

$$|X + 3| \leq \beta_{\mathbb{R}}(M_X, a_X, b_X + 3, a'_X, b'_X)(k)$$

2. multiplication:

$$|3 \times X| \leq \beta_{\mathbb{R}}(M_X, 3 \times a_X, b_X, a'_X, b'_X)(k)$$

3. barycentric mean:

$$\left| \frac{X + Y}{2} \right| \leq \beta_{\mathbb{R}} \left( \frac{M_X + M_Y}{2}, \text{Max}(a_X, a_Y), \frac{b_X + b_Y}{2}, \text{Max}(a'_X, a'_Y), \frac{b'_X + b'_Y}{2} \right) (k)$$

Parametric strategies can be used to transform expressions.

# Projection I

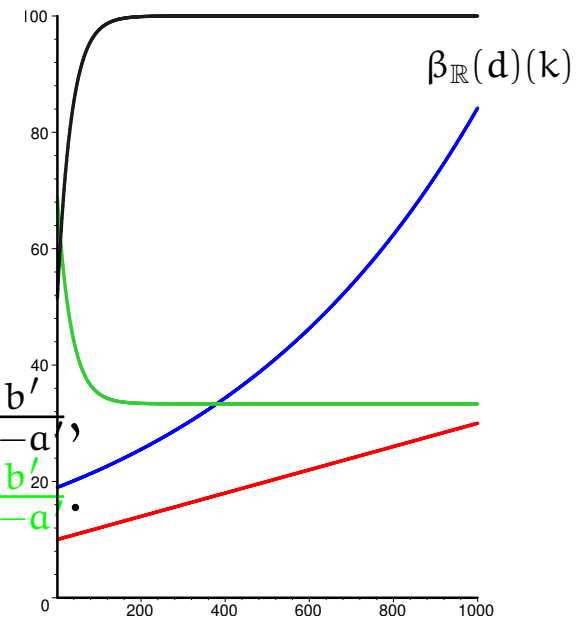
$$\beta_{\mathbb{R}}(M, a, b, a', b')(k) = a \times (M + b' \times k) + b \quad \text{when } a' = 1$$

$$\beta_{\mathbb{R}}(M, a, b, a', b')(k) = a \times \left( (a')^k \times \left( M - \frac{b'}{1-a'} \right) + \frac{b'}{1-a'} \right) + b \quad \text{when } a' \neq 1.$$

Thus, for any  $d \in (\mathbb{R}^+)^5$ ,  
the function  $[k \mapsto \beta_{\mathbb{R}}(d)(k)]$  is:

- either monotonic,
- or anti-monotonic.

$$\left\{ \begin{array}{l} a' > 1, \\ a' = 1, \\ a' < 1 \text{ and } M < \frac{b'}{1-a'}, \\ a' < 1 \text{ and } M > \frac{b'}{1-a'} \end{array} \right.$$



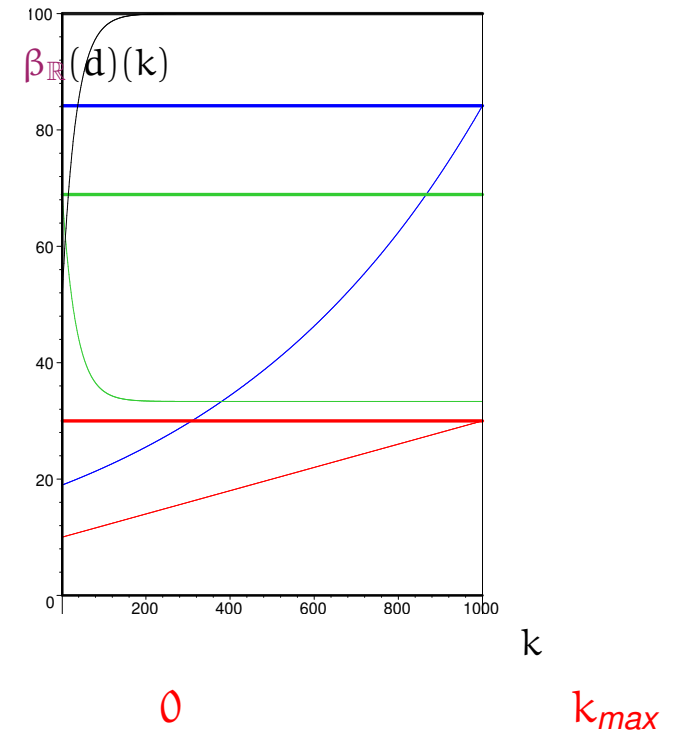
# Projection II

Let  $d \in (\mathbb{R}^+)^5$  and  $k_{max} \in \mathbb{N}$ .

$$bound(d, k_{max}) \triangleq \max(\beta_{\mathbb{R}}(d)(0), \beta_{\mathbb{R}}(d)(k_{max}))$$

For any  $k \in \mathbb{N}$  such that  $0 \leq k \leq k_{max}$ :

$$\beta(d)(k) \leq bound(d, k_{max}).$$



# Incrementing the loop counter

We integrate the current iteration into the first  $k$  iterations:

- the first  $k + 1$  iterations are chosen as the worst case among the first  $k$  iterations and the current iteration;
- the current iteration is reset.

Thus:

$$\text{next}_{\mathbb{R}}(M, a, b, a', b') \triangleq (M, 1, 0, \max(a, a'), \max(b, b')).$$

For any  $k \in \mathbb{N}$ ,  $d \in (\mathbb{R}^+)^5$ ,  $\beta_{\mathbb{R}}(d)(k) \leq \beta_{\mathbb{R}}(\text{next}_{\mathbb{R}}(d))(k + 1)$ .

# About floating point numbers

Floating point numbers occur:

1. in the concrete semantics:

Floating point expressions are translated into real expressions with interval coefficients [Miné—ESOP'04].

In other abstract domains, we handle real numbers.

2. in the abstract domain implementation:

For efficiency purpose, we implement each primitive in floating point arithmetics: each real is safely approximated by an interval with floating point number bounds.



# Overview

1. Introduction
2. Case study
3. Arithmetic-geometric progressions
4. **Benchmarks**
5. Conclusion

# Applications

Arithmetic-geometric progressions provide bounds for :

1. **division by  $\alpha$**  followed by **a multiplication by  $\alpha$** :

⇒ our running example;

2. **barycentric means**:

⇒ at each loop iteration, the value of a variable  $X$  is computed as a barycentric mean of some previous values of  $X$   
(not necessarily the last values);

3. **bounded incremented variables**:

⇒ it replaces the former domain that bounds the difference and the sum between each variable and the loop counter.

# Benchmarks

We analyze three programs in the same family on a **AMD Opteron 248, 8 Gb of RAM** (analyses use only **2 Gb of RAM**).

lines of C	<b>70,000</b>			<b>216,000</b>			<b>379,000</b>		
global variables	13,400			7,500			9,000		
iterations	80	63	<b>37</b>	229	223	<b>53</b>	253	286	<b>74</b>
time/iteration	1mn14s	1mn21s	<b>1mn16s</b>	4mn04s	5mn13s	<b>4mn40s</b>	7mn33s	9mn42s	<b>8mn17s</b>
analysis time	2h18mn	2h05mn	<b>47mn</b>	15h34mn	19h24mn	<b>4h08mn</b>	31h53mn	43h51mn	<b>10h14mn</b>
false alarms	625	24	<b>0</b>	769	64	<b>0</b>	1482	188	<b>0</b>

1. **without using computation time**;
2. with the former **loop counter domain**,  
(without the arithmetic-geometric domain);
3. with **the arithmetic-geometric domain**,  
(without the former loop counter domain).

# Overview

1. Introduction
2. Case study
3. Arithmetic-geometric progressions
4. Benchmarks
5. Conclusion

# A new abstract domain

- non polynomial constraints;
- sound with respect to rounding errors  
(both in the concrete semantics and in the domain implementation);
- accurate  
(we infer bounds on the values that depend on the execution time of the program);
- efficient:
  - in time:  $\mathcal{O}(n \times \ln(n))$  per abstract iteration  
( $n$  denotes the program size),
  - in memory: at most 5 coefficients per variable in the program,
  - sparse implementation.

<http://www.astree.ens.fr>