Certified Roundoff Error Bounds using Semidefinite Programming and Formal Floating Point Arithmetic

Victor Magron, CNRS VERIMAG
Certification is joint work with G. Constantinides and A. Donaldson
Formalization is joint work with T. Weisser and B. Werner

Effective Analysis: Foundations, Implementations, Certification
CIRM, 13 January 2016

Errors and Proofs

Mathematicians and Computer Scientists want to eliminate all the uncertainties on their results. Why?

Errors and Proofs

Mathematicians and Computer Scientists want to eliminate all the uncertainties on their results. Why?

M. Lecat, Erreurs des Mathématiciens des origines à nos jours, 1935. $\rightsquigarrow$ 130 pages of errors! (Euler, Fermat, . . .)

Errors and Proofs

Mathematicians and Computer Scientists want to eliminate all the uncertainties on their results. Why?

M. Lecat, Erreurs des Mathématiciens des origines à nos jours, 1935. $\rightsquigarrow$ 130 pages of errors! (Euler, Fermat, . . .)

Ariane 5 launch failure, Pentium FDIV bug

Errors and Proofs

Mathematicians and Computer Scientists want to eliminate all the uncertainties on their results. Why?

M. Lecat, Erreurs des Mathématiciens des origines à nos jours, 1935. $\leadsto$ 130 pages of errors! (Euler, Fermat, ...)

Ariane 5 launch failure, Pentium FDIV bug

- U.S. Patriot missile killed 28 soldiers from the U.S. Army's
- Internal clock: 0.1 sec intervals
- Roundoff error on the binary constant "0.1"

Errors and Proofs

Guaranteed Optimization

Input : Linear problem  (LP), geometric, semidefinite  (SDP)

Output : solution + **certificate** numeric-symbolic $\rightsquigarrow$ formal

Errors and Proofs

Guaranteed Optimization

Input : Linear problem  (LP), geometric, semidefinite  (SDP)

Output : solution + **certificate** numeric-symbolic $\rightsquigarrow$ formal

Verification of critical systems

Reliable software/hardware embedded codes

Aerospace control

molecular biology, robotics, code synthesis, . . .

Errors and Proofs

Guaranteed Optimization

Input : Linear problem (LP), geometric, semidefinite (SDP)

Output : solution + **certificate** numeric-symbolic $\rightsquigarrow$ formal

Verification of critical systems

Reliable software/hardware embedded codes

Aerospace control

molecular biology, robotics, code synthesis, . . .

Efficient Verification of Nonlinear Systems

- Automated precision tuning of systems/programs
  analysis/synthesis
- Efficiency sparsity correlation patterns
- Certified approximation algorithms

Roundoff Error Bounds

Real : $p(\mathbf{x}) := x_1 \times x_2 + x_3$

Floating-point : $\hat{p}(\mathbf{x}, \mathbf{e}) := [x_1 x_2 (1 + e_1) + x_3](1 + e_2)$

Input variable constraints $\mathbf{x} \in \mathbf{S}$
Finite precision $\rightsquigarrow$ bounds over $\mathbf{e}$

$| e_i | \leqslant 2^{-m}$ $\quad m = 24$ (single) or 53 (double)

Guarantees on absolute round-off error $| \hat{p} - p |$?

Nonlinear Programs

- Polynomials programs : $+, -, \times$

$$x_2x_5 + x_3x_6 + x_1(-x_1 + x_2 + x_3 - x_4 + x_5 + x_6)$$

Nonlinear Programs

- Polynomials programs : $+, -, \times$

$$x_2x_5 + x_3x_6 + x_1(-x_1 + x_2 + x_3 - x_4 + x_5 + x_6)$$

- Semialgebraic programs: $|\cdot|, \sqrt{}, /, \sup, \inf$

$$\frac{4x}{1 + \frac{x}{1.11}}$$

Nonlinear Programs

- Polynomials programs : $+, -, \times$

$$x_2x_5 + x_3x_6 + x_1(-x_1 + x_2 + x_3 - x_4 + x_5 + x_6)$$

- Semialgebraic programs: $|\cdot|, \sqrt{}, /, \sup, \inf$

$$\frac{4x}{1 + \frac{x}{1.11}}$$

- Transcendental programs: $\arctan, \exp, \log, \ldots$

$$\log(1 + \exp(x))$$

Existing Frameworks

Classical methods:

- Abstract domains [Goubault-Putot 11]

  FLUCTUAT: intervals, octagons, zonotopes

- Interval arithmetic [Daumas-Melquiond 10]

  GAPPA: interface with COQ proof assistant

Existing Frameworks

Recent progress:

- Affine arithmetic + SMT [Darulova 14]

  `rosa`: sound compiler for reals (in SCALA)

- Symbolic Taylor expansions [Solovyev 15]

`FPTaylor`: certified optimization (in OCAML and HOL-LIGHT)

Contributions

Maximal Roundoff error of the program implementation of f:

$$r^\star := \max |\hat{f}(\mathbf{x}, \mathbf{e}) - f(\mathbf{x})|$$

Decomposition: **linear** term l w.r.t. $\mathbf{e}$ + nonlinear term h

$$r^\star \leqslant \max |l(\mathbf{x}, \mathbf{e})| + \max |h(\mathbf{x}, \mathbf{e})|$$

- **Semidefinite programming** (SDP) bounds for l
- Coarse bound of h with interval arithmetic

Contributions

1. Comparison with SMT and linear/affine/Taylor arithmetic:

   $\rightsquigarrow$ **Efficient** optimization $\oplus$ **Tight** upper bounds

2. Extensions to transcendental/conditional programs

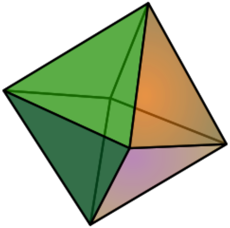
3. Formal verification of SDP bounds

4. Open source tool `Real2Float` (in OCaml and Coq)

What is Semidefinite Programming?

- Linear Programming (LP):

$$\min_{\mathbf{z}} \quad \mathbf{c}^\top \mathbf{z}$$
$$\text{s.t.} \quad \mathbf{A}\,\mathbf{z} \geqslant \mathbf{d} \;.$$



Polyhedron

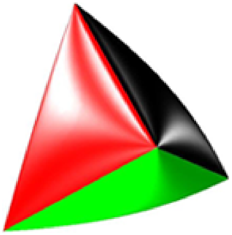
  - Linear cost $\mathbf{c}$
  - Linear inequalities "$\sum_i A_{ij}\, z_j \geqslant d_i$"

What is Semidefinite Programming?

- Semidefinite Programming (SDP):

$$\begin{aligned} \min_{\mathbf{z}} \quad & \mathbf{c}^\top \mathbf{z} \\ \text{s.t.} \quad & \sum_i \mathbf{F}_i z_i \succcurlyeq \mathbf{F}_0 \ . \end{aligned}$$

  - Linear cost $\mathbf{c}$
  - Symmetric matrices $\mathbf{F}_0, \mathbf{F}_i$
  - Linear matrix inequalities "$\mathbf{F} \succcurlyeq 0$" ($\mathbf{F}$ has nonnegative eigenvalues)

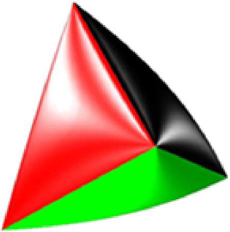


Spectrahedron

What is Semidefinite Programming?

- Semidefinite Programming (SDP):

$$\min_{\mathbf{z}} \quad \mathbf{c}^\top \mathbf{z}$$
$$\text{s.t.} \quad \sum_i \mathbf{F}_i \, z_i \succcurlyeq \mathbf{F}_0 \,, \quad \mathbf{A}\,\mathbf{z} = \mathbf{d} \,.$$



Spectrahedron

- Linear cost $\mathbf{c}$
- Symmetric matrices $\mathbf{F}_0, \mathbf{F}_i$
- Linear matrix inequalities "$\mathbf{F} \succcurlyeq 0$" ($\mathbf{F}$ has nonnegative eigenvalues)

Applications of SDP

- Combinatorial optimization

- Control theory

- Matrix completion

- Unique Games Conjecture (Khot '02) :
  "A *single concrete algorithm* provides **optimal guarantees** among all efficient algorithms for a large class of computational problems."
  (Barak and Steurer survey at ICM'14)

- Solving polynomial optimization (Lasserre '01)

SDP for Polynomial Optimization

- Prove **polynomial inequalities** with SDP:

$$p(a, b) := a^2 - 2ab + b^2 \geqslant 0 \ .$$

- Find $\mathbf{z}$ s.t. $p(a, b) = \begin{pmatrix} a & b \end{pmatrix} \underbrace{\begin{pmatrix} z_1 & z_2 \\ z_2 & z_3 \end{pmatrix}}_{\succcurlyeq 0} \begin{pmatrix} a \\ b \end{pmatrix}$.

- Find $\mathbf{z}$ s.t. $a^2 - 2ab + b^2 = z_1 a^2 + 2 z_2 ab + z_3 b^2 \qquad (\mathbf{A}\,\mathbf{z} = \mathbf{d})$

- $\begin{pmatrix} z_1 & z_2 \\ z_2 & z_3 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}}_{\mathbf{F}_1} z_1 + \underbrace{\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}}_{\mathbf{F}_2} z_2 + \underbrace{\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}}_{\mathbf{F}_3} z_3 \succcurlyeq \underbrace{\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}}_{\mathbf{F}_0}$

SDP for Polynomial Optimization

- Choose a cost $\mathbf{c}$ e.g. $(1,0,1)$ and solve:

$$\begin{aligned} \min_{\mathbf{z}} \quad & \mathbf{c}^{\top} \mathbf{z} \\ \text{s.t.} \quad & \sum_i \mathbf{F}_i \, z_i \succcurlyeq \mathbf{F}_0 \,, \quad \mathbf{A}\,\mathbf{z} = \mathbf{d} \,. \end{aligned}$$

- Solution $\begin{pmatrix} z_1 & z_2 \\ z_2 & z_3 \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \succcurlyeq 0$ (eigenvalues 0 and 2)

- $a^2 - 2ab + b^2 = \begin{pmatrix} a & b \end{pmatrix} \underbrace{\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}}_{\succcurlyeq 0} \begin{pmatrix} a \\ b \end{pmatrix} = (a-b)^2 \,.$

- Solving **SDP** $\Longrightarrow$ Finding **Sums of Squares** certificates

SDP for Polynomial Optimization

General case:

- Semialgebraic set $\mathbf{S} := \{\mathbf{x} \in \mathbb{R}^n : g_1(\mathbf{x}) \geqslant 0, \ldots, g_m(\mathbf{x}) \geqslant 0\}$

- $p^* := \min_{\mathbf{x} \in \mathbf{S}} p(\mathbf{x})$: NP hard

- Sums of squares (SOS) $\Sigma[\mathbf{x}]$ (e.g. $(x_1 - x_2)^2$)

- $\mathcal{Q}(\mathbf{S}) := \left\{ \sigma_0(\mathbf{x}) + \sum_{j=1}^m \sigma_j(\mathbf{x}) g_j(\mathbf{x}), \text{ with } \sigma_j \in \Sigma[\mathbf{x}] \right\}$

- Fix the degree $2k$ of products:

$$\mathcal{Q}_k(\mathbf{S}) := \left\{ \sigma_0(\mathbf{x}) + \sum_{j=1}^{m} \sigma_j(\mathbf{x}) g_j(\mathbf{x}), \text{ with } \deg \sigma_j\, g_j \leqslant 2k \right\}$$

SDP for Polynomial Optimization

- **Hierarchy of SDP relaxations**:
  $\lambda_k := \sup_{\lambda} \Big\{ \lambda : p - \lambda \in \mathcal{Q}_k(\mathbf{S}) \Big\}$

- Convergence guarantees $\lambda_k \uparrow p^*$ [Lasserre 01]

- Can be computed with SDP solvers (CSDP, SDPA)

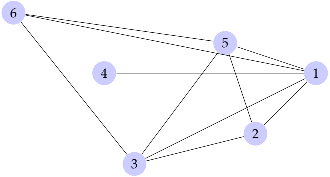
- **"No Free Lunch" Rule**: $\binom{n+2k}{n}$ SDP variables

- Extension to semialgebraic functions $r(\mathbf{x}) = p(\mathbf{x}) / \sqrt{q(\mathbf{x})}$ [Lasserre-Putinar 10]

Sparse SDP Optimization [Waki, Lasserre 06]

- Correlative sparsity pattern (csp) of variables

$$x_2x_5 + x_3x_6 - x_2x_3 - x_5x_6 + x_1(-x_1 + x_2 + x_3 - x_4 + x_5 + x_6)$$



1. Maximal cliques $C_1, \ldots, C_l$
2. Average size $\kappa \rightsquigarrow \binom{\kappa+2k}{\kappa}$ variables

$C_1 := \{1, 4\}$

$C_2 := \{1, 2, 3, 5\}$

$C_3 := \{1, 3, 5, 6\}$

Dense SDP: 210 variables

Sparse SDP: 115 variables

Polynomial Programs

Input: exact $f(\mathbf{x})$, floating-point $\hat{f}(\mathbf{x}, \mathbf{e})$, $\mathbf{x} \in \mathbf{S}$, $\mid e_i \mid \leqslant 2^{-m}$

Output: Bound for $f - \hat{f}$

1: Error $r(\mathbf{x}, \mathbf{e}) := f(\mathbf{x}) - \hat{f}(\mathbf{x}, \mathbf{e}) = \sum_{\boldsymbol{\alpha}} r_{\boldsymbol{\alpha}}(\mathbf{e}) \mathbf{x}^{\boldsymbol{\alpha}}$

2: Decompose $r(\mathbf{x}, \mathbf{e}) = l(\mathbf{x}, \mathbf{e}) + h(\mathbf{x}, \mathbf{e})$, l **linear in** $\mathbf{e}$

3: $l(\mathbf{x}, \mathbf{e}) = \sum_{i=0}^{n'} s_i(\mathbf{x}) e_i$

4: Maximal cliques correspond to $\{\mathbf{x}, e_1\}, \ldots, \{\mathbf{x}, e_{n'}\}$

5: Bound $l(\mathbf{x}, \mathbf{e})$ with sparse SDP relaxations (and h with IA)

Dense relaxation: $\binom{n+n'+2k}{n+n'}$ SDP variables

Sparse relaxation: $n' \binom{n+1+2k}{n+1}$ SDP variables

Preliminary Comparisons

$$f(\mathbf{x}) := x_2x_5 + x_3x_6 - x_2x_3 - x_5x_6 + x_1(-x_1 + x_2 + x_3 - x_4 + x_5 + x_6)$$

$$\mathbf{x} \in [4.00, 6.36]^6\,, \quad \mathbf{e} \in [-\epsilon, \epsilon]^{15}\,, \quad \epsilon = 2^{-24}$$

- Dense SDP: $\binom{6+15+4}{6+15} = 12650$ variables $\rightsquigarrow$ **Out of memory**

Preliminary Comparisons

$$f(\mathbf{x}) := x_2x_5 + x_3x_6 - x_2x_3 - x_5x_6 + x_1(-x_1 + x_2 + x_3 - x_4 + x_5 + x_6)$$

$$\mathbf{x} \in [4.00, 6.36]^6\,, \quad \mathbf{e} \in [-\epsilon, \epsilon]^{15}\,, \quad \epsilon = 2^{-24}$$

- Dense SDP: $\binom{6+15+4}{6+15} = 12650$ variables $\leadsto$ **Out of memory**
- Sparse SDP `Real2Float` tool: $15\binom{6+1+4}{6+1} = 4950 \leadsto 759\epsilon$

Preliminary Comparisons

$$f(\mathbf{x}) := x_2x_5 + x_3x_6 - x_2x_3 - x_5x_6 + x_1(-x_1 + x_2 + x_3 - x_4 + x_5 + x_6)$$

$$\mathbf{x} \in [4.00, 6.36]^6 \,, \quad \mathbf{e} \in [-\epsilon, \epsilon]^{15} \,, \quad \epsilon = 2^{-24}$$

- Dense SDP: $\binom{6+15+4}{6+15} = 12650$ variables $\rightsquigarrow$ **Out of memory**
- Sparse SDP `Real2Float` tool: $15\binom{6+1+4}{6+1} = 4950 \rightsquigarrow 759\epsilon$
- Interval arithmetic: 922ϵ ($10 \times$ less CPU)

Preliminary Comparisons

$$f(\mathbf{x}) := x_2x_5 + x_3x_6 - x_2x_3 - x_5x_6 + x_1(-x_1 + x_2 + x_3 - x_4 + x_5 + x_6)$$

$$\mathbf{x} \in [4.00, 6.36]^6 , \quad \mathbf{e} \in [-\epsilon, \epsilon]^{15} , \quad \epsilon = 2^{-24}$$

- Dense SDP: $\binom{6+15+4}{6+15} = 12650$ variables $\leadsto$ **Out of memory**
- Sparse SDP `Real2Float` tool: $15\binom{6+1+4}{6+1} = 4950 \leadsto 759\epsilon$
- Interval arithmetic: 922ϵ ($10 \times$ less CPU)
- Symbolic Taylor `FPTaylor` tool: 721ϵ ($21 \times$ more CPU)

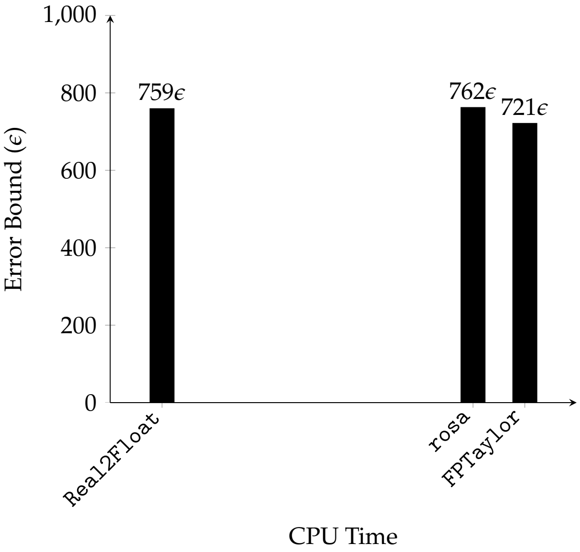
Preliminary Comparisons

$$f(\mathbf{x}) := x_2x_5 + x_3x_6 - x_2x_3 - x_5x_6 + x_1(-x_1 + x_2 + x_3 - x_4 + x_5 + x_6)$$

$$\mathbf{x} \in [4.00, 6.36]^6 \,, \quad \mathbf{e} \in [-\epsilon, \epsilon]^{15} \,, \quad \epsilon = 2^{-24}$$

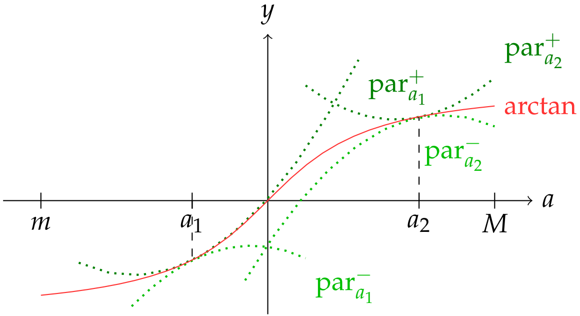
- Dense SDP: $\binom{6+15+4}{6+15} = 12650$ variables $\leadsto$ **Out of memory**
- Sparse SDP `Real2Float` tool: $15\binom{6+1+4}{6+1} = 4950 \leadsto 759\epsilon$
- Interval arithmetic: 922ϵ ($10 \times$ less CPU)
- Symbolic Taylor `FPTaylor` tool: 721ϵ ($21 \times$ more CPU)
- SMT-based `rosa` tool: 762ϵ ($19 \times$ more CPU)

Preliminary Comparisons



Extensions: Transcendental Programs

Reduce $f^* := \inf_{\mathbf{x} \in \mathbf{K}} f(\mathbf{x})$ to semialgebraic optimization



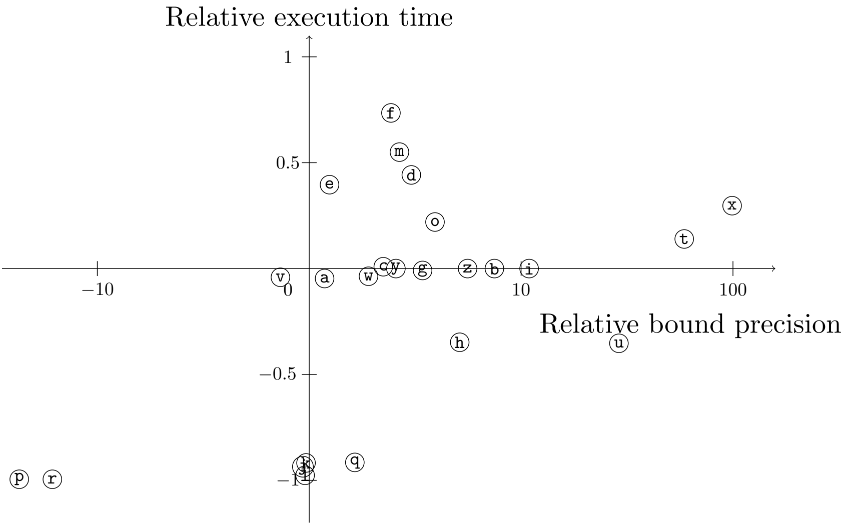
Extensions: Programs with Conditionals

`if` $(p(\mathbf{x}) \leqslant 0)\ f(\mathbf{x});$ `else` $g(\mathbf{x});$

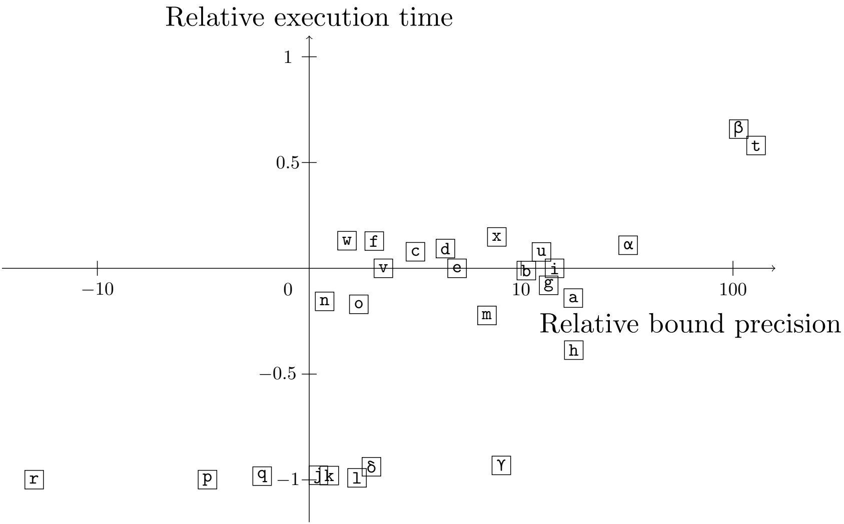
DIVERGENCE PATH ERROR:

$$
\begin{aligned}
r^\star := \max\{ & \\
& \max_{p(\mathbf{x}) \leqslant 0, p(\mathbf{x},\mathbf{e}) \geqslant 0} |\, \hat{f}(\mathbf{x},\mathbf{e}) - g(\mathbf{x}) \,| \\
& \max_{p(\mathbf{x}) \geqslant 0, p(\mathbf{x},\mathbf{e}) \leqslant 0} |\, \hat{g}(\mathbf{x},\mathbf{e}) - f(\mathbf{x}) \,| \\
& \max_{p(\mathbf{x}) \geqslant 0, p(\mathbf{x},\mathbf{e}) \geqslant 0} |\, \hat{f}(\mathbf{x},\mathbf{e}) - f(\mathbf{x}) \,| \\
& \max_{p(\mathbf{x}) \leqslant 0, p(\mathbf{x},\mathbf{e}) \leqslant 0} |\, \hat{g}(\mathbf{x},\mathbf{e}) - g(\mathbf{x}) \,| \\
\} &
\end{aligned}
$$

Comparison with rosa



Comparison with FPTaylor



Interval Coefficient Polynomials

SOS certificate for $0 \leqslant x_1, x_2 \leqslant 1 \land x_1^2 \leqslant x_2 \Rightarrow x_2 - 2x_1 + 1 \geqslant 0$:
$x_2 - 2x_1 + 1 = (1 - x_1)^2 + x_2 - x_1^2$

Interval Coefficient Polynomials

SOS certificate for $0 \leqslant x_1, x_2 \leqslant 1 \wedge x_1^2 \leqslant x_2 \Rightarrow x_2 - 2x_1 + 1 \geqslant 0$:
$x_2 - 2x_1 + 1 = (1 - x_1)^2 + x_2 - x_1^2$

SDP solvers only find **approximate** certificates:

$$\begin{aligned} x_2 - 2x_1 + 1 \quad \simeq \quad & 1.00007(0.99977 - 1.00022x_1 - 0.00011x_2)^2 \\ + \quad & 0.000332(-0.408035x_1 + 0.816664x_2 - 0.408126)^2 \\ + \quad & 0.000284x_2 + 0.000116(1 - x_2) + 1.00034(x_2 - x_1^2) \end{aligned}$$

Interval Coefficient Polynomials

SOS certificate for $0 \leqslant x_1, x_2 \leqslant 1 \wedge x_1^2 \leqslant x_2 \Rightarrow x_2 - 2x_1 + 1 \geqslant 0$:
$x_2 - 2x_1 + 1 = (1 - x_1)^2 + x_2 - x_1^2$

SDP solvers only find **approximate** certificates:

$$\begin{aligned} x_2 - 2x_1 + 1 \quad \simeq \quad & 1.00007(0.99977 - 1.00022x_1 - 0.00011x_2)^2 \\ + \quad & 0.000332(-0.408035x_1 + 0.816664x_2 - 0.408126)^2 \\ + \quad & 0.000284x_2 + 0.000116(1 - x_2) + 1.00034(x_2 - x_1^2) \end{aligned}$$

Exact error polynomial:

$$\begin{aligned} \varepsilon(\mathbf{x}) \quad := \quad & 0.000232209x_1^2 - 5.81334 \times 10^{-7}x_1x_2 - 0.0000297356x_1 \\ + \quad & 0.000221436x_2^2 + 0.0000621035x_2 - 0.000201126 \end{aligned}$$

How to employ **numerical** certificates for formal verification?

How to use numerical certificates in Coq?

tactic	strategy

$$\varepsilon(\mathbf{x}) := 0.000232209x_1^2 - 5.81334 \times 10^{-7}x_1x_2 - 0.0000297356x_1 + 0.000221436x_2^2 + 0.0000621035x_2 - 0.000201126$$

How to use numerical certificates in COQ?

tactic	strategy
MICROMEGA	uses heuristics to get an exact representation

$$\varepsilon'(\mathbf{x}) \quad = \quad 0$$

How to use numerical certificates in Coq?

tactic	strategy
Micromega	uses heuristics to get an exact representation
NLCertify	gives lower bound on ε by exact computations

$$\begin{aligned} \varepsilon^* \quad := \quad & 0.000232209 x_1^2 - 5.81334 \times 10^{-7} x_1 x_2 - 0.0000297356 x_1 \\ + \quad & 0.000221436 x_2^2 + 0.0000621035 x_2 - 0.000201126 \end{aligned}$$

How to use numerical certificates in Coq?

tactic	strategy
Micromega	uses heuristics to get an exact representation
NLCertify	gives lower bound on ε by exact computations
NLVerify	use interval arithmetics to bound ε

$$\varepsilon^* \quad := \quad \text{interval enclosure of } \varepsilon$$

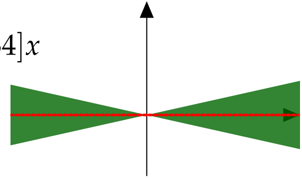
Interval Coefficient Polynomials

- Floating point numbers $\mathbb{F}_{(p)} := \mathbb{F}_{r,p}$ with radix r and precision p
  $\leadsto$ Fast, certified inside COQ (FLOCQ, Boldo/Melquiond).
  In this talk $r = 10$, in the implementation $r = 2$.
- Intervals $\mathbb{I}_p := \mathbb{I}_{r,p}$ with floating point bounds $\mathbb{F}_p$
  $\leadsto$ Keep track of roundoff errors.
- Box constraints: $\mathbf{x} \in \mathbf{B}$
- Coefficient enclosure $[\cdot]_p$ and variable enclosure $|\cdot|_{\mathbf{B}}$

Interval Coefficient Polynomials

Replace coefficients by intervals to speed up computation:

$$
\begin{aligned}
f &:= \frac{1}{3}x - \frac{1}{3}x = 0 \\
[f]_2 &= [0.33, 0.34]x - [0.33, 0.34]x \\
[0]_2 &= [0.00, 0.00]
\end{aligned}
$$



Interval Coefficient Polynomials

Replace variables by intervals to obtain bounds on the function:
With $\mathbf{B} = [-1, 1] \times [0, 1] \times [0, 1]$,

$$\begin{aligned} x_1(x_2 - x_3) &= x_1 x_2 - x_2 x_3 \\ |x_1(x_2 - x_3)|_{\mathbf{B}} &= [-1, 1][-1, 1] = [-1, 1] \\ |x_1 x_2 - x_1 x_3|_{\mathbf{B}} &= [-1, 1] - [-1, 1] = [-2, 2] \end{aligned}$$

Coq Implementation

Theorem:

$$\Big| [f]_p \Big|_{\mathbf{B}} \subseteq [\ell, \infty) \Rightarrow f \geqslant \ell \text{ on } \mathbf{B}.$$

Coq Implementation

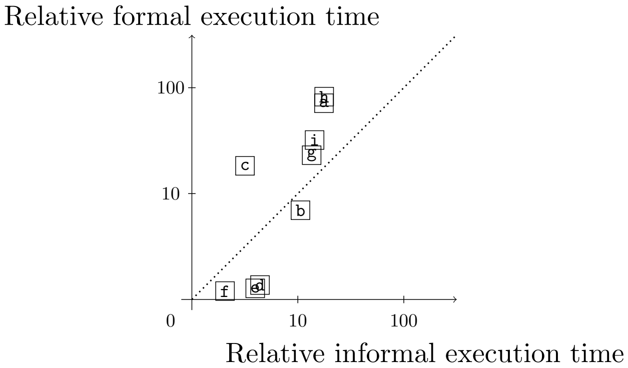
Theorem:

$$\left| [f]_p \right|_{\mathbf{B}} \subseteq [\ell, \infty) \Rightarrow f \geqslant \ell \text{ on } \mathbf{B}.$$

COQVersion:

```
Lemma toPolI_ok p box x :
x ∈ box → eval x p ∈ Vencl box (toPolI p).
```

Comparison with FPTaylor



Conclusion

Sparse SDP relaxations analyze **NONLINEAR PROGRAMS**:

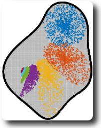
- Polynomial and transcendental programs
- Handles conditionals, input uncertainties, ...
- Certified $\rightsquigarrow$ Formal roundoff error bounds
- `Real2Float` open source tool:

  `http://nl-certify.forge.ocamlcore.org/real2float.html`

Conclusion

Further research:

- Improve formal polynomial checker
- Roundoff error analysis with `while`/`for` loops



- Automatic **FPGA** code generation

End

Thank you for your attention!

http://www-verimag.imag.fr/~magron

- V. Magron, G. Constantinides, A. Donaldson. Certified Roundoff Error Bounds Using Semidefinite Programming, arxiv.org/abs/1507.03331