Certified Roundoff Error Bounds using Semidefinite Programming

Victor Magron, CNRS VERIMAG

joint work with G. Constantinides and A. Donaldson

INRIA Mescal Team Seminar
19 November 2015
Mathematicians and Computer Scientists want to eliminate all the uncertainties on their results. Why?
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.

130 pages of errors! (Euler, Fermat, Sylvester, …)
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.

130 pages of errors! (Euler, Fermat, Sylvester, ...)

Ariane 5 launch failure, Pentium FDIV bug
Errors and Proofs

GUARANTEED OPTIMIZATION

Input: Linear problem \( \text{LP} \), geometric, semidefinite \( \text{SDP} \)

Output: solution + **certificate** numeric-symbolic \( \sim \) formal
Errors and Proofs

Guaranteed Optimization
Input: Linear problem (LP), geometric, semidefinite (SDP)
Output: solution + certificate numeric-symbolic formal

Verification of critical systems
Reliable software/hardware embedded codes
Aerospace control
molecular biology, robotics, code synthesis, ...
Guaranteed Optimization

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

Output: solution + certificate numeric-symbolic 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
Rounding Error Bounds

Real: \( p(x) := x_1 \times x_2 + x_3 \)

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

Input variable uncertainties \( x \in S \)
Finite precision \( \sim \) bounds over \( e \)

\[ |e_i| \leq 2^{-m} \quad m = 24 \text{ (single)} \text{ or } 53 \text{ (double)} \]

Guarantees on absolute round-off error \( |\hat{p} - p| \) ?
Nonlinear Programs

- **Polynomials programs**: $\pm, -, \times$

\[ x_2 x_5 + x_3 x_6 + x_1 (-x_1 + x_2 + x_3 - x_4 + x_5 + x_6) \]
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]
  \textsc{FLUCTUAT}: intervals, octagons, zonotopes
- Interval arithmetic [Daumas-Melquiond 10]
  \textsc{GAPPA}: interface with \textsc{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 Rounding error of the program implementation of $f$:

\[ r^* := \max |\hat{f}(x, e) - f(x)| \]

Decomposition: linear term $l$ w.r.t. $e$ + nonlinear term $h$

\[ r^* \leq \max |l(x, e)| + \max |h(x, e)| \]

- Sparse SDP bounds for $l$
- Coarse bound of $h$ with interval arithmetic
Contributions

1. **Comparison** with SMT and linear/affine arithmetic:
   - More **Efficient** optimization  
   - **Tight** upper bounds

2. Extensions to **transcendental**/conditional programs

3. Formal verification of SDP bounds

4. Open source tool **Real2Float** (in OCAML and COQ)
Introduction

Semidefinite Programming for Polynomial Optimization

Rounding Error Bounds with Sparse SDP

Conclusion
What is Semidefinite Programming?

- Linear Programming (LP):
  \[
  \min_{\mathbf{z}} \quad \mathbf{c}^\top \mathbf{z}
  \]
  \[
  \text{s.t.} \quad \mathbf{A} \mathbf{z} \geq \mathbf{d}.
  \]

- Linear cost \( \mathbf{c} \)

- Linear inequalities \( \sum_i A_{ij} z_j \geq d_i \)
What is Semidefinite Programming?

Semidefinite Programming (SDP):

\[
\min_z \quad c^\top z \\
\text{s.t.} \quad \sum_i F_i z_i \succeq F_0.
\]

- Linear cost \( c \)
- Symmetric matrices \( F_0, F_i \)
- Linear matrix inequalities "\( F \succeq 0 \)" (\( F \) has nonnegative eigenvalues)
What is Semidefinite Programming?

- Semidefinite Programming (SDP):

  \[
  \min_{z} \quad c^\top z \\
  \text{s.t.} \quad \sum_{i} F_i z_i \succeq F_0, \quad A z = d .
  \]

- Linear cost \( c \)

- Symmetric matrices \( F_0, F_i \)

- Linear matrix inequalities "\( F \succeq 0"\) (\( F \) 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:

\[ p(a, b) := a^2 - 2ab + b^2 \geq 0. \]

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

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

\[
\begin{pmatrix} z_1 & z_2 \\ z_2 & z_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} z_1 + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} z_2 + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} z_3 \geq \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}
\]
Choose a cost $c$ e.g. $(1, 0, 1)$ and solve:

$$\min_z c^\top z \quad \text{s.t.} \quad \sum_i F_i z_i \succeq F_0 , \quad A z = d .$$

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

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

Solving SDP $\implies$ Finding Sums of Squares certificates
General case:

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

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

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

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

- Fix the degree $2k$ of sums of squares
  $Q_k(S) := Q(S) \cap \mathbb{R}_{2k}[x]$
SDP for Polynomial Optimization

- Hierarchy of SDP relaxations:
  \[ \lambda_k := \sup_{\lambda} \left\{ \lambda : p - \lambda \in Q_k(S) \right\} \]

- 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 \( r(x) = p(x) / \sqrt{q(x)} \) [Lasserre-Putinar 10]
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 \( C_1, \ldots, C_l \)

\[
\begin{align*}
C_1 & := \{1, 4\} \\
C_2 & := \{1, 2, 3, 5\} \\
C_3 & := \{1, 3, 5, 6\}
\end{align*}
\]

2. Average size \( \kappa \sim (\frac{\kappa + 2\kappa}{\kappa}) \) variables

Dense SDP: 210 variables
Sparse SDP: 115 variables
Introduction

Semidefinite Programming for Polynomial Optimization

**Rounding Error Bounds with Sparse SDP**

Conclusion
Polynomial Programs

**Input:** exact $f(x)$, floating-point $\hat{f}(x, e)$, $x \in S$, $|e_i| \leq 2^{-m}$

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

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

2. Decompose $r(x, e) = l(x, e) + h(x, e)$, $l$ linear in $e$

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

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

5. Bound $l(x, e)$ 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

\[ f(x) := x_2x_5 + x_3x_6 - x_2x_3 - x_5x_6 + x_1(-x_1 + x_2 + x_3 - x_4 + x_5 + x_6) \]

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

- **Dense SDP**: \( \binom{6+15+4}{6+15} = 12650 \) variables \( \leadsto \) **Out of memory**

- **Sparse SDP** Real2Float tool: \( 15\binom{6+1+4}{6+1} = 4950 \) \( \leadsto 789\epsilon \)

- **Interval arithmetic**: \( 2023\epsilon \) (17 \times less CPU)

- **Symbolic Taylor** FPTaylor tool: \( 936\epsilon \) (16.3 \times more CPU)

- **SMT-based** rosa tool: \( 789\epsilon \) (4.6 \times more CPU)
Preliminary Comparisons

CPU Time

Real2Float  |  rosa  |  FPTaylor
---|---|---
789$\epsilon$  |  789$\epsilon$  |  936$\epsilon$

Error Bound ($\epsilon$)
Extensions: Transcendental Programs

Given $K$ a compact set and $f$ a transcendental function, bound

$$f^* = \inf_{x \in K} f(x)$$

and prove $f^* \geq 0$

- $f$ is under-approximated by a semialgebraic function $f_{sa}$

- Reduce the problem $f_{sa}^* := \inf_{x \in K} f_{sa}(x)$ to a polynomial optimization problem (POP)
Approximation theory: Chebyshev/Taylor models

- mandatory for non-polynomial problems

- hard to combine with Sum-of-Squares techniques (degree of approximation)
Maxplus Approximations

- Initially introduced to solve Optimal Control Problems [Fleming-McEneaney 00]
- **Curse of dimensionality** reduction [McEaneney Kluberg, Gaubert-McEneaney-Qu 11, Qu 13]. Allowed to solve instances of dim up to 15 (inaccessible by grid methods)
- In our context: approximate **transcendental** functions
Maxplus Approximations

**Definition**

Let $\gamma \geq 0$. A function $\phi : \mathbb{R}^n \to \mathbb{R}$ is said to be $\gamma$-semiconvex if the function $x \mapsto \phi(x) + \frac{\gamma}{2} \|x\|_2^2$ is convex.
Exact parsimonious maxplus representations
Maxplus Approximations

Exact parsimonious maxplus representations

\[ y \]

\[ a \]
if \( p(x) \leq 0 \) \( f(x) \); else \( g(x) \);

**Divergence path error:**

\[
r^* := \max \left\{ \begin{align*}
\max_{p(x) \leq 0, p(x,e) \geq 0} |\hat{f}(x,e) - g(x)| \\
\max_{p(x) \geq 0, p(x,e) \leq 0} |\hat{g}(x,e) - f(x)| \\
\max_{p(x) \geq 0, p(x,e) \geq 0} |\hat{f}(x,e) - f(x)| \\
\max_{p(x) \leq 0, p(x,e) \leq 0} |\hat{g}(x,e) - g(x)|
\end{align*} \right\}
\]
Benchmarks

Doppler

\[ u \in [-100, 100] \]
\[ v \in [20, 20000] \]
\[ T \in [-30, 50] \]

\[
\text{let } t_1 = 331.4 + 0.6T \quad \text{in} \quad \frac{-t_1v}{(t_1 + u)^2}
\]
Benchmarks

Kepler2

\[ x \in [4, 6.36]^6 \]
\[ x_1x_4(-x_1 + x_2 + x_3 - x_4 + x_5 + x_6) + x_2x_5(x_1 - x_2 + x_3 + x_4 - x_5 + x_6) + x_3x_6(x_1 + x_2 - x_3 + x_4 + x_5 - x_6) - x_2x_3x_4 - x_1x_3x_5 - x_1x_2x_6 - x_4x_5x_6 \]
**Benchmarks**

\[ x \in [0.1, 0.3] \]

\[ \frac{4x}{1 + \frac{x}{1.11}} \]

Verhulst

CPU Time

<table>
<thead>
<tr>
<th></th>
<th>Error Bound ((\epsilon))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real2Float</td>
<td>3.07(\epsilon), 3.16(\epsilon)</td>
</tr>
<tr>
<td>FPTaylor</td>
<td>6.15(\epsilon)</td>
</tr>
<tr>
<td>Rosa</td>
<td>6.15(\epsilon)</td>
</tr>
</tbody>
</table>
Benchmarks

logexp

\( x \in [-8, 8] \)

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

Semidefinite Programming for Polynomial Optimization

Rounding Error Bounds with Sparse SDP

Conclusion
Conclusion

Sparse SDP relaxations analyze NONLINEAR PROGRAMS:

- **Polynomial** and **transcendental** programs
- Handles conditionals, input uncertainties, …
- Certified 🐪 $\sim$ Formal 🧮 rounding error bounds
- Real2Float open source tool:
  
  https://forge.ocamlcore.org/projects/nl-certify
Conclusion

Further research:

- Improve formal polynomial checker

- Alternative polynomial bounds using geometric programming (T. de Wolff, S. Iliman)

- Mixed linear/SDP certificates (trade-off CPU/precision)

- More program verification: while loops

- Automatic FPGA code generation
Thank you for your attention!

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