

Enclosures of Roundoff Errors using SDP

Victor Magron, CNRS



Jointly Certified Upper Bounds with **G. Constantinides** and **A. Donaldson**

Analysis and Synthesis of Floating-point Programs
Dagstuhl Seminar, August 2017



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

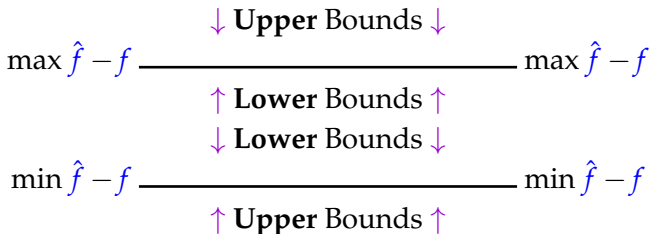
Real : $f(\mathbf{x}) := x_1 \times x_2 + x_3$

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

Input variable constraints $\mathbf{x} \in \mathbf{X}$

Finite precision \rightsquigarrow bounds over $\mathbf{e} \in \mathbf{E}$: $|e_i| \leq 2^{-53}$ (double)

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



Nonlinear Programs

- **Polynomials** programs : +, -, ×

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

Nonlinear Programs

- **Polynomials** programs : +, -, ×

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

- **Semialgebraic** programs: |·|, √, /, sup, inf

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

Nonlinear Programs

- **Polynomials** programs : +, -, ×

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

- **Semialgebraic** programs: | · |, √, /, sup, inf

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

- **Transcendental** programs: arctan, exp, log, ...

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

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], [Darulova 17]
 rosa: sound compiler for reals (SCALA)
- Symbolic Taylor expansions [Solovyev 15], [Chiang 17]
 FPTaylor: certified optimization (OCAML/HOL-LIGHT)
- Guided random testing s3fp [Chiang 14]

Contributions

Maximal Roundoff error of the program implementation of f :

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

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

$$\max |l| + \max |h| \geq r^* \geq \max |l| - \max |h|$$

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

Contributions

Maximal Roundoff error of the program implementation of f :

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

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

$$\max |l| + \max |h| \geq r^* \geq \max |l| - \max |h|$$

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

↓ **Upper Bounds** ↓

↑ **Lower Bounds** ↑

↓ **Lower Bounds** ↓

↑ **Upper Bounds** ↑

Sparse SDP relaxations

Robust SDP relaxations

Contributions

- 1 General **SDP** framework for **upper** and **lower** bounds
- 2 **Comparison** with SMT & affine/Taylor arithmetic:
 \rightsquigarrow **Efficient** optimization \oplus **Tight** upper bounds
- 3 Extensions to **transcendental**/conditional programs
- 4 Formal verification of SDP bounds 🐔
- 5 Open source tools:

Real2Float (in OCAML and COQ)

↓ Upper Bounds ↓

↑ Upper Bounds ↑

FPSPD (in MATLAB)

↑ Lower Bounds ↑
↓ Lower Bounds ↓

Introduction

Semidefinite Programming for Polynomial Optimization

Upper Bounds with Sparse SDP

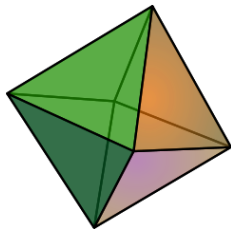
Lower Bounds with Robust SDP

Conclusion

What is Semidefinite Programming?

- Linear Programming (LP):

$$\begin{aligned} \min_{\mathbf{z}} \quad & \mathbf{c}^\top \mathbf{z} \\ \text{s.t.} \quad & \mathbf{A} \mathbf{z} \geq \mathbf{d} . \end{aligned}$$



- Linear cost \mathbf{c}
- Linear inequalities “ $\sum_i A_{ij} z_j \geq d_i$ ”

Polyhedron

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):

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

- 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

SDP for Polynomial Optimization

- Prove **polynomial inequalities** with SDP:

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

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

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

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

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)

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

- Solving **SDP** \implies Finding **SUMS OF SQUARES** certificates

SDP for Polynomial Optimization

NP hard General Problem: $f^* := \min_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x})$

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

SDP for Polynomial Optimization

NP hard General Problem: $f^* := \min_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x})$

■ Semialgebraic set $\mathbf{X} := \{\mathbf{x} \in \mathbb{R}^n : g_1(\mathbf{x}) \geq 0, \dots, g_m(\mathbf{x}) \geq 0\}$

■ $:= [0, 1]^2 = \{\mathbf{x} \in \mathbb{R}^2 : x_1(1 - x_1) \geq 0, \quad x_2(1 - x_2) \geq 0\}$

SDP for Polynomial Optimization

NP hard General Problem: $f^* := \min_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x})$

■ Semialgebraic set $\mathbf{X} := \{\mathbf{x} \in \mathbb{R}^n : g_1(\mathbf{x}) \geq 0, \dots, g_m(\mathbf{x}) \geq 0\}$

■ $:= [0, 1]^2 = \{\mathbf{x} \in \mathbb{R}^2 : x_1(1 - x_1) \geq 0, \quad x_2(1 - x_2) \geq 0\}$

$$\underbrace{x_1 x_2}_f + \frac{1}{8} = \frac{1}{2} \overbrace{\left(x_1 + x_2 - \frac{1}{2}\right)^2}^{\sigma_0} + \frac{1}{2} \overbrace{x_1(1 - x_1)}^{\sigma_1} + \frac{1}{2} \overbrace{x_2(1 - x_2)}^{\sigma_2}$$

SDP for Polynomial Optimization

NP hard General Problem: $f^* := \min_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x})$

- Semialgebraic set $\mathbf{X} := \{\mathbf{x} \in \mathbb{R}^n : g_1(\mathbf{x}) \geq 0, \dots, g_m(\mathbf{x}) \geq 0\}$
- $:= [0, 1]^2 = \{\mathbf{x} \in \mathbb{R}^2 : x_1(1 - x_1) \geq 0, \quad x_2(1 - x_2) \geq 0\}$

$$\underbrace{x_1 x_2}_f + \frac{1}{8} = \frac{1}{2} \overbrace{\left(x_1 + x_2 - \frac{1}{2}\right)^2}^{\sigma_0} + \frac{1}{2} \overbrace{x_1(1 - x_1)}^{\sigma_1} + \frac{1}{2} \overbrace{x_2(1 - x_2)}^{\sigma_2}$$

- Sums of squares (SOS) σ_i

SDP for Polynomial Optimization

NP hard General Problem: $f^* := \min_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x})$

- Semialgebraic set $\mathbf{X} := \{\mathbf{x} \in \mathbb{R}^n : g_1(\mathbf{x}) \geq 0, \dots, g_m(\mathbf{x}) \geq 0\}$
- $:= [0, 1]^2 = \{\mathbf{x} \in \mathbb{R}^2 : x_1(1 - x_1) \geq 0, \quad x_2(1 - x_2) \geq 0\}$

$$\underbrace{x_1 x_2}_f + \frac{1}{8} = \frac{1}{2} \overbrace{\left(x_1 + x_2 - \frac{1}{2}\right)^2}^{\sigma_0} + \frac{1}{2} \overbrace{x_1(1 - x_1)}^{\sigma_1} + \frac{1}{2} \overbrace{x_2(1 - x_2)}^{\sigma_2}$$

- Sums of squares (SOS) σ_i

SDP for Polynomial Optimization

NP hard General Problem: $f^* := \min_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x})$

- Semialgebraic set $\mathbf{X} := \{\mathbf{x} \in \mathbb{R}^n : g_1(\mathbf{x}) \geq 0, \dots, g_m(\mathbf{x}) \geq 0\}$
- $:= [0, 1]^2 = \{\mathbf{x} \in \mathbb{R}^2 : x_1(1 - x_1) \geq 0, \quad x_2(1 - x_2) \geq 0\}$

$$\underbrace{f}_{x_1 x_2} + \frac{1}{8} = \overbrace{\frac{1}{2} \left(x_1 + x_2 - \frac{1}{2} \right)^2}^{\sigma_0} + \overbrace{\frac{1}{2} x_1(1 - x_1)}^{\sigma_1} + \overbrace{\frac{1}{2} x_2(1 - x_2)}^{\sigma_2}$$

- Sums of squares (SOS) σ_i
- Bounded degree:

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

SDP for Polynomial Optimization

- **Hierarchy of SDP relaxations:**

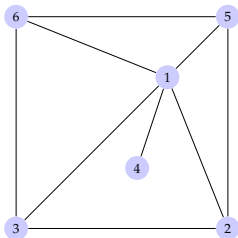
$$\lambda_k := \sup_{\lambda} \left\{ \lambda : f - \lambda \in \mathcal{Q}_k(\mathbf{X}) \right\}$$

- Convergence guarantees $\lambda_k \uparrow f^*$ [Lasserre 01]
- Can be computed with SDP solvers (CSDP, SDPA)
- **“No Free Lunch” Rule:** $\binom{n+2k}{n}$ SDP variables

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, \dots, 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

Introduction

Semidefinite Programming for Polynomial Optimization

Upper Bounds with Sparse SDP

Lower Bounds with Robust SDP

Conclusion

Polynomial Programs

Input: exact $f(\mathbf{x})$, approx $\hat{f}(\mathbf{x}, \mathbf{e})$, $\mathbf{x} \in \mathbf{X}$, $|e_i| \leq 2^{-53}$

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

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

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

- 3: $l(\mathbf{x}, \mathbf{e}) = \sum_{i=1}^m s_i(\mathbf{x}) e_i$

- 4: Maximal cliques correspond to $\{\mathbf{x}, e_1\}, \dots, \{\mathbf{x}, e_m\}$

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

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

Preliminary Comparisons

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

$$\mathbf{x} \in [4.00, 6.36]^6, \quad \mathbf{e} \in [-\epsilon, \epsilon]^{15}, \quad \epsilon = 2^{-53}$$

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

Preliminary Comparisons

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

$$\mathbf{x} \in [4.00, 6.36]^6, \quad \mathbf{e} \in [-\epsilon, \epsilon]^{15}, \quad \epsilon = 2^{-53}$$

- **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

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

$$\mathbf{x} \in [4.00, 6.36]^6, \quad \mathbf{e} \in [-\epsilon, \epsilon]^{15}, \quad \epsilon = 2^{-53}$$

- **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)

Preliminary Comparisons

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

$$\mathbf{x} \in [4.00, 6.36]^6, \quad \mathbf{e} \in [-\epsilon, \epsilon]^{15}, \quad \epsilon = 2^{-53}$$

- **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)

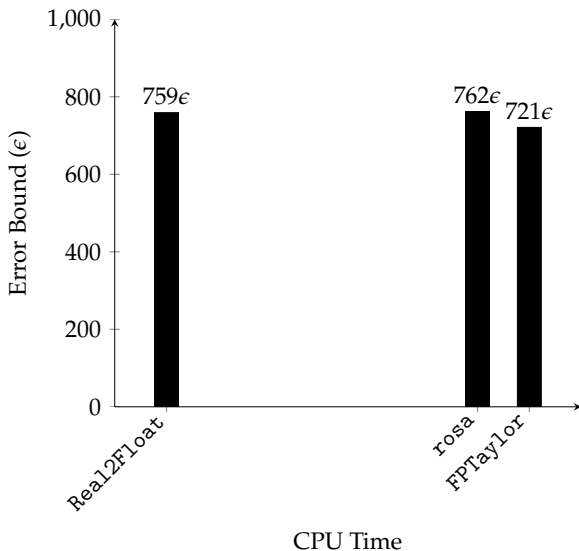
Preliminary Comparisons

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

$$\mathbf{x} \in [4.00, 6.36]^6, \quad \mathbf{e} \in [-\epsilon, \epsilon]^{15}, \quad \epsilon = 2^{-53}$$

- **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



Introduction

Semidefinite Programming for Polynomial Optimization

Upper Bounds with Sparse SDP

Lower Bounds with Robust SDP

Conclusion

Method 1: geneig [Lasserre 11]

Generalized eigenvalue problem:

$$f^* := \min_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x}) \leq \lambda_k := \sup_{\lambda} \lambda$$

s.t. $\mathbf{M}_k(f \mathbf{y}) \succeq \lambda \mathbf{M}_k(\mathbf{y})$.

Method 1: geneig [Lasserre 11]

Generalized eigenvalue problem:

$$f^* := \min_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x}) \leq \lambda_k := \sup_{\lambda} \lambda$$

s.t. $\mathbf{M}_k(f \mathbf{y}) \succeq \lambda \mathbf{M}_k(\mathbf{y})$.

Uniform distribution moments: $\mathbf{y}_\alpha := \int_{\mathbf{X}} \mathbf{x}^\alpha d\mathbf{x}$

Localizing matrices $\mathbf{M}_k(f \mathbf{y})$:

$$\mathbf{M}_1(f \mathbf{y}) = \begin{matrix} & \begin{matrix} 1 & x_1 & x_2 \end{matrix} \\ \begin{matrix} 1 \\ x_1 \\ x_2 \end{matrix} & \begin{pmatrix} \int_{\mathbf{X}} f(\mathbf{x}) d\mathbf{x} & \int_{\mathbf{X}} f(\mathbf{x}) x_1 d\mathbf{x} & \int_{\mathbf{X}} f(\mathbf{x}) x_2 d\mathbf{x} \\ \int_{\mathbf{X}} f(\mathbf{x}) x_1 d\mathbf{x} & \int_{\mathbf{X}} f(\mathbf{x}) x_1^2 d\mathbf{x} & \int_{\mathbf{X}} f(\mathbf{x}) x_1 x_2 d\mathbf{x} \\ \int_{\mathbf{X}} f(\mathbf{x}) x_2 d\mathbf{x} & \int_{\mathbf{X}} f(\mathbf{x}) x_2 x_1 d\mathbf{x} & \int_{\mathbf{X}} f(\mathbf{x}) x_2^2 d\mathbf{x} \end{pmatrix} \end{matrix}$$

Method 1: geneig [Lasserre 11]

Generalized eigenvalue problem:

$$f^* := \min_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x}) \leq \lambda_k := \sup_{\lambda} \lambda$$

s.t. $\mathbf{M}_k(f \mathbf{y}) \succeq \lambda \mathbf{M}_k(\mathbf{y})$.

Uniform distribution moments: $\mathbf{y}_\alpha := \int_{\mathbf{X}} \mathbf{x}^\alpha d\mathbf{x}$

Localizing matrices $\mathbf{M}_k(f \mathbf{y})$:

$$\mathbf{M}_1(f \mathbf{y}) = \begin{matrix} & \begin{matrix} 1 & x_1 & x_2 \end{matrix} \\ \begin{matrix} 1 \\ x_1 \\ x_2 \end{matrix} & \begin{pmatrix} \int_{\mathbf{X}} f(\mathbf{x}) d\mathbf{x} & \int_{\mathbf{X}} f(\mathbf{x}) x_1 d\mathbf{x} & \int_{\mathbf{X}} f(\mathbf{x}) x_2 d\mathbf{x} \\ \int_{\mathbf{X}} f(\mathbf{x}) x_1 d\mathbf{x} & \int_{\mathbf{X}} f(\mathbf{x}) x_1^2 d\mathbf{x} & \int_{\mathbf{X}} f(\mathbf{x}) x_1 x_2 d\mathbf{x} \\ \int_{\mathbf{X}} f(\mathbf{x}) x_2 d\mathbf{x} & \int_{\mathbf{X}} f(\mathbf{x}) x_2 x_1 d\mathbf{x} & \int_{\mathbf{X}} f(\mathbf{x}) x_2^2 d\mathbf{x} \end{pmatrix} \end{matrix}$$

Theorem [Lasserre 11, de Klerk et al. 17]

$$\lambda_k \downarrow f^* \quad \text{and} \quad \lambda_k - f^* = \mathcal{O}(1/\sqrt{k})$$

Elementary calculation with $f(\mathbf{x}) = \sum_{\alpha} f_{\alpha} \mathbf{x}^{\alpha}$:

$$f^* := \min_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x}) \leq f_k^H := \min_{|\eta + \beta| \leq 2k} \sum_{\alpha} f_{\alpha} \frac{\gamma_{\eta + \alpha, \beta}}{\gamma_{\eta, \beta}}$$

Method 2: mvbeta [DeKlerk et al. 17]

Elementary calculation with $f(\mathbf{x}) = \sum_{\alpha} f_{\alpha} \mathbf{x}^{\alpha}$:

$$f^* := \min_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x}) \leq f_k^H := \min_{|\eta + \beta| \leq 2k} \sum_{\alpha} f_{\alpha} \frac{\gamma_{\eta + \alpha, \beta}}{\gamma_{\eta, \beta}}$$

Multivariate beta distribution moments:

$$\gamma_{\eta, \beta} := \int_{\mathbf{X}} \mathbf{x}^{\eta} (1 - \mathbf{x})^{\beta} d\mathbf{x}.$$

Method 2: mvbeta [DeKlerk et al. 17]

Elementary calculation with $f(\mathbf{x}) = \sum_{\alpha} f_{\alpha} \mathbf{x}^{\alpha}$:

$$f^* := \min_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x}) \leq f_k^H := \min_{|\eta+\beta| \leq 2k} \sum_{\alpha} f_{\alpha} \frac{\gamma_{\eta+\alpha, \beta}}{\gamma_{\eta, \beta}}$$

Multivariate beta distribution moments:

$$\gamma_{\eta, \beta} := \int_{\mathbf{X}} \mathbf{x}^{\eta} (1 - \mathbf{x})^{\beta} d\mathbf{x}.$$

Theorem [DeKlerk et al. 17]

$$f_k^H \downarrow f^* \quad \text{and} \quad \lambda_k - f^* = \mathcal{O}(1/\sqrt{k})$$

Method 3: robustsdp

Robust SDP with $l(\mathbf{x}, \mathbf{e}) = \sum_{i=1}^m s_i(\mathbf{x})e_i$:

$$l^* := \min_{(\mathbf{x}, \mathbf{e}) \in \mathbf{X} \times \mathbf{E}} l(\mathbf{x}, \mathbf{e}) \leq \lambda'_k := \sup_{\lambda} \lambda$$

$$\text{s.t. } \forall \mathbf{e} \in \mathbf{E}, \mathbf{M}_k(l \mathbf{y}) \succeq \lambda \mathbf{M}_k(\mathbf{y}).$$

Method 3: robustsdp

Robust SDP with $l(\mathbf{x}, \mathbf{e}) = \sum_{i=1}^m s_i(\mathbf{x})e_i$:

$$l^* := \min_{(\mathbf{x}, \mathbf{e}) \in \mathbf{X} \times \mathbf{E}} l(\mathbf{x}, \mathbf{e}) \leq \lambda'_k := \sup_{\lambda} \lambda$$

$$\text{s.t. } \forall \mathbf{e} \in \mathbf{E}, \mathbf{M}_k(l \mathbf{y}) \succeq \lambda \mathbf{M}_k(\mathbf{y}).$$

$$\text{Linearity} \rightsquigarrow \mathbf{M}_k(l \mathbf{y}) = \sum_{i=1}^m e_i \mathbf{M}_k(s_i \mathbf{y})$$

$$\text{Factorization} \rightsquigarrow \mathbf{M}_k(s_i \mathbf{y}) = \mathbf{L}_k^i \mathbf{R}_k^i$$

Method 3: robustsdp

Robust SDP with $l(\mathbf{x}, \mathbf{e}) = \sum_{i=1}^m s_i(\mathbf{x})e_i$:

$$l^* := \min_{(\mathbf{x}, \mathbf{e}) \in \mathbf{X} \times \mathbf{E}} l(\mathbf{x}, \mathbf{e}) \leq \lambda'_k := \sup_{\lambda} \lambda$$

$$\text{s.t. } \forall \mathbf{e} \in \mathbf{E}, \mathbf{M}_k(l \mathbf{y}) \succcurlyeq \lambda \mathbf{M}_k(\mathbf{y}).$$

$$\text{Linearity } \rightsquigarrow \mathbf{M}_k(l \mathbf{y}) = \sum_{i=1}^m e_i \mathbf{M}_k(s_i \mathbf{y})$$

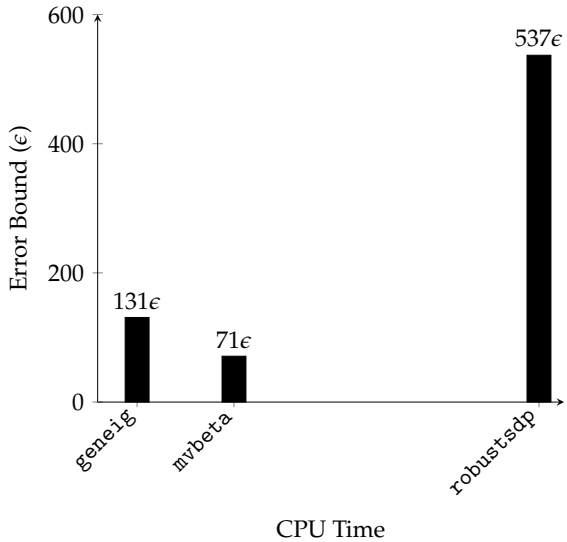
$$\text{Factorization } \rightsquigarrow \mathbf{M}_k(s_i \mathbf{y}) = \mathbf{L}_k^i \mathbf{R}_k^i$$

Theorem following from [El Ghaoui et al. 98]

$$\lambda'_k \downarrow l^* \text{ and } \lambda'_k = \sup_{\lambda, \mathbf{S}, \mathbf{G}} \lambda$$

$$\text{s.t. } \begin{pmatrix} -\lambda \mathbf{M}_k(\mathbf{y}) - \mathbf{L}_k \mathbf{S} \mathbf{L}_k^T & \mathbf{R}_k^T + \mathbf{L}_k \mathbf{G} \\ \mathbf{R}_k - \mathbf{G} \mathbf{L}_k^T & \mathbf{S} \end{pmatrix} \succcurlyeq 0, \\ \mathbf{S}^T = \mathbf{S}, \mathbf{G}^T = -\mathbf{G}.$$

Benchmark kepler0 with $k = 2$



Introduction

Semidefinite Programming for Polynomial Optimization

Upper Bounds with Sparse SDP



Lower Bounds with Robust SDP

Conclusion

Conclusion

Sparse/Robust SDP relaxations for NONLINEAR PROGRAMS:

- Polynomial and transcendental programs

- Certified  \rightsquigarrow Formal  roundoff error bounds
(Joint work with T. Weisser and B. Werner)

- Real2Float and FPSDP open source tools:

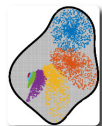
<http://nl-certify.forge.ocamlcore.org/real2float.html>

<https://github.com/magronv/FPSDP>

Conclusion

Further research:

- Automatic **FPGA** code generation
- Handling `while/for` loops



Master / PhD Positions Available !



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, *ACM Trans. Math. Softw.*, 2017.

↓ Upper Bounds ↓

↑ Upper Bounds ↑

- V. Magron. Interval Enclosures of Upper Bounds of Roundoff Errors using Semidefinite Programming, arxiv.org/abs/1611.01318.

↑ Lower Bounds ↑

↓ Lower Bounds ↓
