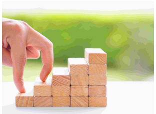
Certified polynomial optimization

Victor Magron, LAAS CNRS

MINT Summer School “Moments, Positive Polynomials and their Applications”   28 June 2022

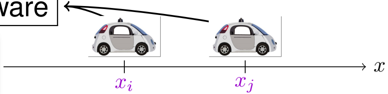


Motivation: verification of nonlinear systems

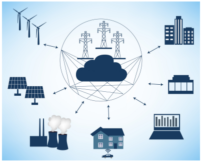
SAFETY of critical parts for **computing** $\oplus$ **physical** devices

Control Software/Hardware

Cars

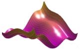


Smart Grids



Space Systems

... CAST AS CERTIFIED OPTIMIZATION $\rightsquigarrow$ SOLVE **OFFLINE**

Input: linear  semidefinite  polynomial 

Output: value + numerical/symbolic/formal **certificate**

Motivation: match with symbolic/formal tools

Positivity certificates

- Stability proofs of critical control systems (Lyapunov)

Motivation: match with symbolic/formal tools

Positivity certificates

- Stability proofs of critical control systems (Lyapunov)

- Certified function evaluation [Chevillard et. al 11]

Motivation: match with symbolic/formal tools

Positivity certificates

- Stability proofs of critical control systems (Lyapunov)

- Certified function evaluation [Chevillard et. al 11]

COQ HOL-LIGHT

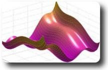
Kepler's conjecture (1611): the max density of sphere packings is $\pi/\sqrt{18}$

Fly**sp**eck : **F**ormalizing the **p**roof of **K**epler [Hales et al. 94]
Certification of thousands of "tight" nonlinear inequalities [Hales et al. 17]

Conic programs & polynomial optimization

NP-hard NON CONVEX Problem $f^\star = \inf f(\mathbf{x})$

Theory

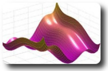
(Primal)		(Dual)
$\inf \int f \, d\mu$		$\sup \lambda$
with μ proba $\Rightarrow$	**INFINITE LP**	$\Leftarrow$ with $f - \lambda \geqslant 0$

Conic programs & polynomial optimization

NP-hard NON CONVEX Problem $f^\star = \inf f(\mathbf{x})$

Practice

(Primal **Relaxation**)



(Dual **Strengthening**)

moments $\int \mathbf{x}^\alpha \, d\mu$

$f - \lambda =$ **sum of squares**

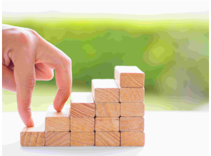
finite number $\Rightarrow$ **SDP** $\Leftarrow$ **fixed** degree

LASSERRE'S HIERARCHY of **CONVEX PROBLEMS** $\uparrow f^*$

[Lasserre/Parrilo 01]

degree d & n vars $\Longrightarrow \binom{n+2d}{n}$ **SDP** VARIABLES

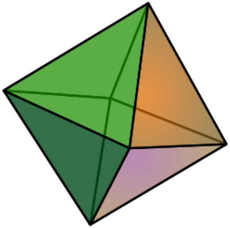
Numeric solvers $\Longrightarrow$ **Approx** Certificate



Conic Programming: LP

- Linear Programming (LP):

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



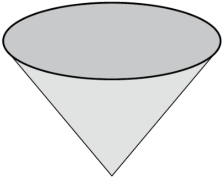
Polyhedron

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

Conic Programming: SOCP

- Second-order Cone Programming (SOCP):

$$\min_{\mathbf{z}} \quad \mathbf{c}^\top \mathbf{z}$$
$$\text{s.t.} \quad z_1 \geq \sqrt{{z_2}^2 + \cdots + {z_n}^2}$$



Second-order Cone

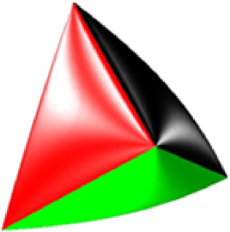
  - Linear cost $\mathbf{c}$
  - convex set defined by the linear/quadratic inequalities

Conic Programming: SDP

- Semidefinite Programming (SDP):

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

  - 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

Deciding nonnegativity

$\mathbf{x} = (x_1, \ldots, x_n)$ | **co-NP hard problem: check** $f \geqslant 0$ **on** $\mathbf{K}$

$f \in \mathbb{Q}[\mathbf{x}]$ | **NP hard problem:** $\min\{f(\mathbf{x}) : \mathbf{x} \in \mathbf{K}\}$

Deciding nonnegativity

$\mathbf{x} = (x_1, \ldots, x_n)$ **co-NP hard problem: check $f \geqslant 0$ on $\mathbf{K}$**

$f \in \mathbb{Q}[\mathbf{x}]$ **NP hard problem:** $\min\{f(\mathbf{x}) : \mathbf{x} \in \mathbf{K}\}$

1. Unconstrained $\rightsquigarrow \mathbf{K} = \mathbb{R}^n$

2. Constrained $\rightsquigarrow \mathbf{K} = \{\mathbf{x} \in \mathbb{R}^n : g_j(\mathbf{x}) \geqslant 0\}$ $\quad g_j \in \mathbb{Q}[\mathbf{x}]$

$\deg f, \deg g_j \leqslant d$

Deciding nonnegativity

$\mathbf{x} = (x_1, \ldots, x_n)$ **co-NP hard problem: check $f \geqslant 0$ on $\mathbf{K}$**

$f \in \mathbb{Q}[\mathbf{x}]$ **NP hard problem:** $\min\{f(\mathbf{x}) : \mathbf{x} \in \mathbf{K}\}$

1. Unconstrained $\rightsquigarrow$ $\mathbf{K} = \mathbb{R}^n$

2. Constrained $\rightsquigarrow$ $\mathbf{K} = \{\mathbf{x} \in \mathbb{R}^n : g_j(\mathbf{x}) \geqslant 0\}$ $\quad g_j \in \mathbb{Q}[\mathbf{x}]$

$\deg f, \deg g_j \leqslant d$

[Collins 75] CAD **doubly exp. in n poly. in d**

[Grigoriev-Vorobjov 88, Basu-Pollack-Roy 98, Safey El Din-Schost 03]

Critical points **singly exponential time** $(m+1)\,\tau\, d^{\mathcal{O}(n)}$

Deciding nonnegativity

Sums of squares (SOS)

$\sigma = {h_1}^2 + \cdots + {h_p}^2$

Deciding nonnegativity

Sums of squares (SOS)

$\sigma = {h_1}^2 + \cdots + {h_p}^2$

HILBERT 17TH PROBLEM: f SOS of rational functions?

[Artin 27] **Yes**!

Deciding nonnegativity

Sums of squares (SOS)

$\sigma = {h_1}^2 + \cdots + {h_p}^2$

HILBERT 17TH PROBLEM: f SOS of rational functions?

[Artin 27] **Yes**!

[Lasserre/Parrilo 01] **Numerical** solvers compute σ

Semidefinite programming (SDP) $\rightsquigarrow$ **approximate** certificates

Deciding nonnegativity

Sums of squares (SOS)

$\sigma = {h_1}^2 + \cdots + {h_p}^2$

Hilbert 17th Problem: f SOS of rational functions?

[Artin 27] **Yes**!

[Lasserre/Parrilo 01] **Numerical** solvers compute σ

Semidefinite programming (SDP) $\rightsquigarrow$ **approximate** certificates

$$\boxed{\simeq \quad \rightarrow \quad =}$$

The Question of Exact Certification

How to go from **approximate** to **exact** certification?

Decomposing nonnegative polynomials

1. **Reznick**'s representation $f = \frac{\sigma}{(x_1^2 + \cdots + x_n^2)^D}$
   positive definite form f
   [Reznick 95]

2. **Hilbert-Artin**'s representation $f = \frac{\sigma}{h^2}$
   $f \geqslant 0$
   [Artin 27]

3. **Putinar**'s representation $f = \sigma_0 + \sigma_1 \, g_1 + \cdots + \sigma_m \, g_m$
   $f > 0$ on compact K $\deg \sigma_i \leqslant 2D$
   [Putinar 93]

Decomposing nonnegative polynomials

- Deciding **polynomial nonnegativity**

$$f(a,b) = a^2 - 2ab + b^2 \geqslant 0$$

- $f(a,b) = \begin{pmatrix} a & b \end{pmatrix} \underbrace{\begin{pmatrix} z_1 & z_2 \\ z_2 & z_3 \end{pmatrix}}_{\succcurlyeq 0} \begin{pmatrix} a \\ b \end{pmatrix}$

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

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

Decomposing nonnegative polynomials

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

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

- 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)

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

- SUMS OF SQUARES certificates via **SDP**

Decomposing nonnegative polynomials

4. **Circuit** polynomial
   $f = b_1 \mathbf{x}^{\boldsymbol{\alpha}(1)} + \cdots + b_t \mathbf{x}^{\boldsymbol{\alpha}(t)} + b_{\boldsymbol{\beta}} \mathbf{x}^{\boldsymbol{\beta}}$

   $b_j > 0 \quad \boldsymbol{\alpha}(j) \in (2\mathbb{N})^n$
   $\boldsymbol{\beta} = \lambda_1 \boldsymbol{\alpha}(1) + \cdots + \lambda_t \boldsymbol{\alpha}(t) \quad \lambda_j > 0$ and $\lambda_1 + \cdots + \lambda_t = 1$

Decomposing nonnegative polynomials

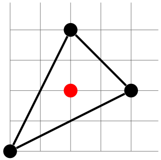
4. **Circuit** polynomial

$f = b_1 \mathbf{x}^{\boldsymbol{\alpha}(1)} + \cdots + b_t \mathbf{x}^{\boldsymbol{\alpha}(t)} + b_{\boldsymbol{\beta}} \mathbf{x}^{\boldsymbol{\beta}}$

$b_j > 0 \quad \boldsymbol{\alpha}(j) \in (2\mathbb{N})^n$

$\boldsymbol{\beta} = \lambda_1 \boldsymbol{\alpha}(1) + \cdots + \lambda_t \boldsymbol{\alpha}(t) \quad \lambda_j > 0$ and $\lambda_1 + \cdots + \lambda_t = 1$

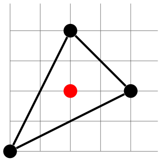
$f = 1 + x_1^2 x_2^4 + x_1^4 x_2^2 - 3 x_1^2 x_2^2$



Decomposing nonnegative polynomials

4. **Circuit** polynomial
   $f = b_1 \mathbf{x}^{\boldsymbol{\alpha}(1)} + \cdots + b_t \mathbf{x}^{\boldsymbol{\alpha}(t)} + b_{\boldsymbol{\beta}} \mathbf{x}^{\boldsymbol{\beta}}$

   $b_j > 0 \quad \boldsymbol{\alpha}(j) \in (2\mathbb{N})^n$
   $\boldsymbol{\beta} = \lambda_1 \boldsymbol{\alpha}(1) + \cdots + \lambda_t \boldsymbol{\alpha}(t) \quad \lambda_j > 0$ and $\lambda_1 + \cdots + \lambda_t = 1$



$$f = 1 + x_1^2 x_2^4 + x_1^4 x_2^2 - 3 x_1^2 x_2^2$$

Circuit number $\Theta_f = \prod_{j=1}^{t} \left(\frac{b_j}{\lambda_j} \right)^{\lambda_j}$

Decomposing nonnegative polynomials

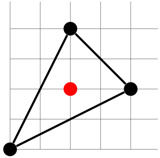
4 **Circuit** polynomial

$f = b_1 \mathbf{x}^{\boldsymbol{\alpha}(1)} + \cdots + b_t \mathbf{x}^{\boldsymbol{\alpha}(t)} + b_{\boldsymbol{\beta}} \mathbf{x}^{\boldsymbol{\beta}}$

$b_j > 0 \quad \boldsymbol{\alpha}(j) \in (2\mathbb{N})^n$

$\boldsymbol{\beta} = \lambda_1 \boldsymbol{\alpha}(1) + \cdots + \lambda_t \boldsymbol{\alpha}(t) \quad \lambda_j > 0$ and $\lambda_1 + \cdots + \lambda_t = 1$

$f = 1 + x_1^2 x_2^4 + x_1^4 x_2^2 - 3 x_1^2 x_2^2$



Circuit number $\Theta_f = \prod_{j=1}^t \left(\frac{b_j}{\lambda_j} \right)^{\lambda_j}$

Theorem [Illiman-de Wolff 16]

$f \geqslant 0 \Leftrightarrow |b_{\boldsymbol{\beta}}| \leqslant \Theta_f$ or ($b_{\boldsymbol{\beta}} \geqslant -\Theta_f$, $\boldsymbol{\beta}$ even)

sums of nonnegative circuits computed via geometric programs

Decomposing nonnegative polynomials

5 **arithmetic-geometric-mean-exponential (AGE)**

$f = c_1 \exp[\mathbf{x} \cdot \boldsymbol{\alpha}(1)] + \cdots + c_t \exp[\mathbf{x} \cdot \boldsymbol{\alpha}(t)] + \beta \exp[\mathbf{x} \cdot \boldsymbol{\alpha}(0)]$

$c_j \in \mathbb{Q}_{>0} \quad \beta \in \mathbb{Q} \quad \boldsymbol{\alpha}(j) \in \mathbb{N}^n$

Decomposing nonnegative polynomials

5 **arithmetic-geometric-mean-exponential (AGE)**

$f = c_1 \exp[\mathbf{x} \cdot \boldsymbol{\alpha}(1)] + \cdots + c_t \exp[\mathbf{x} \cdot \boldsymbol{\alpha}(t)] + \beta \exp[\mathbf{x} \cdot \boldsymbol{\alpha}(0)]$

$c_j \in \mathbb{Q}_{>0} \quad \beta \in \mathbb{Q} \quad \boldsymbol{\alpha}(j) \in \mathbb{N}^n$

relative entropy $D(\boldsymbol{\nu}, \mathbf{c}) = \sum_j \nu_j \log \frac{\nu_j}{c_j}$

Decomposing nonnegative polynomials

5 **arithmetic-geometric-mean-exponential (AGE)**

$f = c_1 \exp[\mathbf{x} \cdot \boldsymbol{\alpha}(1)] + \cdots + c_t \exp[\mathbf{x} \cdot \boldsymbol{\alpha}(t)] + \beta \exp[\mathbf{x} \cdot \boldsymbol{\alpha}(0)]$

$c_j \in \mathbb{Q}_{>0} \quad \beta \in \mathbb{Q} \quad \boldsymbol{\alpha}(j) \in \mathbb{N}^n$

relative entropy $D(\boldsymbol{\nu}, \mathbf{c}) = \sum_j \nu_j \log \frac{\nu_j}{c_j}$

Theorem (Chandrasekaran-Shah 16)

$f \geqslant 0 \Leftrightarrow \exists \boldsymbol{\nu} \mid D(\boldsymbol{\nu}, \mathbf{c}) \leqslant \beta$ and $\sum_j \boldsymbol{\alpha}(j) \nu_j = (\mathbf{1} \cdot \boldsymbol{\nu}) \boldsymbol{\alpha}(0)$

SAGE (sums of AGE) computed via relative entropy programs

From approximate to exact certificates

Approximate solutions

sum of squares of $a^2 - 2ab + b^2$?

$(1.00001a - 0.99998b)^2$!

$$a^2 - 2ab + b^2 \simeq (1.00001a - 0.99998b)^2$$

$$a^2 - 2ab + b^2 \neq 1.0000200001a^2 - 1.9999799996ab + 0.9999600004b^2$$

$\boxed{\simeq \quad \to \quad =}$?

Rational SOS decompositions

- Let $f \in \mathbb{R}[x]$ and $f \geqslant 0$ on $\mathbb{R}$ ($n = 1$)

Theorem

There exist $f_1, f_2 \in \mathbb{R}[x]$ s.t. $f = {f_1}^2 + {f_2}^2$.

Rational SOS decompositions

- Let $f \in \mathbb{R}[x]$ and $f \geqslant 0$ on $\mathbb{R}$ ($n = 1$)

Theorem

There exist $f_1, f_2 \in \mathbb{R}[x]$ s.t. $f = {f_1}^2 + {f_2}^2$.

Proof.

$f = h^2(q + ir)(q - ir)$ □

Rational SOS decompositions

- Let $f \in \mathbb{R}[x]$ and $f \geqslant 0$ on $\mathbb{R}$ ($n = 1$)

Theorem

There exist $f_1, f_2 \in \mathbb{R}[x]$ s.t. $f = {f_1}^2 + {f_2}^2$.

Proof.

$f = h^2(q + ir)(q - ir)$ □

Examples

$$1 + x + x^2 = \left(x + \frac{1}{2}\right)^2 + \left(\frac{\sqrt{3}}{2}\right)^2$$

$$1 + x + x^2 + x^3 + x^4 = \left(x^2 + \frac{1}{2}x + \frac{1 + \sqrt{5}}{4}\right)^2 + \left(\frac{\sqrt{10 + 2\sqrt{5}} + \sqrt{10 - 2\sqrt{5}}}{4}x + \frac{\sqrt{10 - 2\sqrt{5}}}{4}\right)^2$$

Rational SOS decompositions

- $f \in \mathbb{Q}[x] \cap \mathring{\Sigma}[x]$ (interior of the SOS cone)

Existence Question

Does there exist $f_i \in \mathbb{Q}[x], c_i \in \mathbb{Q}^{>0}$ s.t. $f = \sum_i c_i f_i^2$?

Rational SOS decompositions

- $f \in \mathbb{Q}[x] \cap \mathring{\Sigma}[x]$ (interior of the SOS cone)

Existence Question

Does there exist $f_i \in \mathbb{Q}[x], c_i \in \mathbb{Q}^{>0}$ s.t. $f = \sum_i c_i f_i^2$?

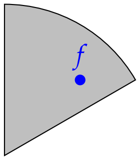
Examples

$$1 + x + x^2 = \left(x + \frac{1}{2}\right)^2 + \left(\frac{\sqrt{3}}{2}\right)^2 = 1\left(x + \frac{1}{2}\right)^2 + \frac{3}{4}(1)^2$$

$$1 + x + x^2 + x^3 + x^4 = \left(x^2 + \frac{1}{2}x + \frac{1+\sqrt{5}}{4}\right)^2 + \left(\frac{\sqrt{10+2\sqrt{5}} + \sqrt{10-2\sqrt{5}}}{4}x + \frac{\sqrt{10-2\sqrt{5}}}{4}\right)^2 = ???$$

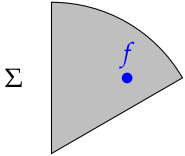
Round & Project algorithm [Peyrl-Parrilo 08]

Σ



$f \in \mathring{\Sigma}[\mathbf{x}]$ with $\deg f = 2D$

Round & Project algorithm [Peyrl-Parrilo 08]



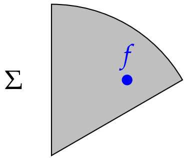
$f \in \mathring{\Sigma}[\mathbf{x}]$ with $\deg f = 2D$

Find $\tilde{\mathbf{G}}$ with SDP at tolerance $\tilde{\delta}$ satisfying

$f(\mathbf{x}) \simeq \mathbf{v}_D{}^T(\mathbf{x})\, \tilde{\mathbf{G}}\, \mathbf{v}_D(\mathbf{x}) \quad \tilde{\mathbf{G}} \succ 0$

$\mathbf{v}_D(\mathbf{x})$: vector of monomials of $\deg \leqslant D$

Round & Project algorithm [Peyrl-Parrilo 08]



$f \in \mathring{\Sigma}[\mathbf{x}]$ with $\deg f = 2D$

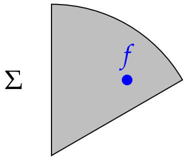
Find $\tilde{\mathrm{G}}$ with SDP at tolerance $\tilde{\delta}$ satisfying

$f(\mathbf{x}) \simeq \mathbf{v}_D{}^T(\mathbf{x})\, \tilde{\mathrm{G}}\, \mathbf{v}_D(\mathbf{x}) \quad \tilde{\mathrm{G}} \succ 0$

$\mathbf{v}_D(\mathbf{x})$: vector of monomials of $\deg \leqslant D$

Exact $G \implies f_\gamma = \sum_{\alpha'+\beta'=\gamma} G_{\alpha',\beta'}$

Round & Project algorithm [Peyrl-Parrilo 08]



$f \in \mathring{\Sigma}[\mathbf{x}]$ with $\deg f = 2D$

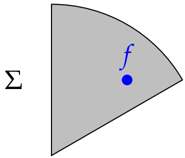
Find $\tilde{\mathbf{G}}$ with SDP at tolerance $\tilde{\delta}$ satisfying

$f(\mathbf{x}) \simeq \mathbf{v}_D{}^T(\mathbf{x})\, \tilde{\mathbf{G}}\, \mathbf{v}_D(\mathbf{x}) \quad \tilde{\mathbf{G}} \succ 0$

$\mathbf{v}_D(\mathbf{x})$: vector of monomials of $\deg \leqslant D$

Exact $G \implies f_\gamma = \sum_{\alpha'+\beta'=\gamma} G_{\alpha',\beta'} \quad f_{\alpha+\beta} = \sum_{\alpha'+\beta'=\alpha+\beta} G_{\alpha',\beta'}$

Round & Project algorithm [Peyrl-Parrilo 08]



$f \in \mathring{\Sigma}[\mathbf{x}]$ with $\deg f = 2D$

Find $\tilde{\mathbf{G}}$ with SDP at tolerance $\tilde{\delta}$ satisfying

$f(\mathbf{x}) \simeq \mathbf{v}_D{}^T(\mathbf{x})\, \tilde{\mathbf{G}}\, \mathbf{v}_D(\mathbf{x}) \quad \tilde{\mathbf{G}} \succ 0$

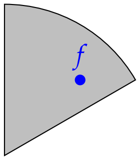
$\mathbf{v}_D(\mathbf{x})$: vector of monomials of $\deg \leqslant D$

Exact $G \implies f_\gamma = \sum_{\alpha'+\beta'=\gamma} G_{\alpha',\beta'} \quad f_{\alpha+\beta} = \sum_{\alpha'+\beta'=\alpha+\beta} G_{\alpha',\beta'}$

1. **Rounding step** $\hat{G} \leftarrow \texttt{round}\left(\tilde{G}, \hat{\delta}\right)$

Round & Project algorithm [Peyrl-Parrilo 08]

Σ



$f \in \mathring{\Sigma}[\mathbf{x}]$ with $\deg f = 2D$

Find $\tilde{G}$ with SDP at tolerance $\tilde{\delta}$ satisfying

$f(\mathbf{x}) \simeq \mathbf{v}_D{}^T(\mathbf{x})\, \tilde{G}\, \mathbf{v}_D(\mathbf{x}) \quad \tilde{G} \succ 0$

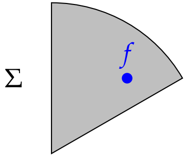
$\mathbf{v}_D(\mathbf{x})$: vector of monomials of $\deg \leqslant D$

Exact $G \implies f_\gamma = \sum_{\alpha'+\beta'=\gamma} G_{\alpha',\beta'} \quad f_{\alpha+\beta} = \sum_{\alpha'+\beta'=\alpha+\beta} G_{\alpha',\beta'}$

1. **Rounding step** $\hat{G} \leftarrow \texttt{round}\left(\tilde{G}, \hat{\delta}\right)$
2. **Projection step**

   $G_{\alpha,\beta} \leftarrow \hat{G}_{\alpha,\beta} - \frac{1}{\eta(\alpha+\beta)} \left(\sum_{\alpha'+\beta'=\alpha+\beta} \hat{G}_{\alpha',\beta'} - f_{\alpha+\beta}\right)$

Round & Project algorithm [Peyrl-Parrilo 08]



$f \in \mathring{\Sigma}[\mathbf{x}]$ with $\deg f = 2D$

Find $\tilde{G}$ with SDP at tolerance $\tilde{\delta}$ satisfying

$f(\mathbf{x}) \simeq \mathbf{v}_D{}^T(\mathbf{x})\, \tilde{G}\, \mathbf{v}_D(\mathbf{x}) \quad \tilde{G} \succ 0$

$\mathbf{v}_D(\mathbf{x})$: vector of monomials of $\deg \leqslant D$

Exact $G \implies f_\gamma = \sum_{\alpha'+\beta'=\gamma} G_{\alpha',\beta'} \quad f_{\alpha+\beta} = \sum_{\alpha'+\beta'=\alpha+\beta} G_{\alpha',\beta'}$

1. **Rounding step** $\hat{G} \leftarrow \texttt{round}\left(\tilde{G}, \hat{\delta}\right)$
2. **Projection step**
   $G_{\alpha,\beta} \leftarrow \hat{G}_{\alpha,\beta} - \frac{1}{\eta(\alpha+\beta)}\left(\sum_{\alpha'+\beta'=\alpha+\beta} \hat{G}_{\alpha',\beta'} - f_{\alpha+\beta}\right)$

Small enough $\tilde{\delta}, \hat{\delta} \implies f(\mathbf{x}) = \mathbf{v}_D{}^T(\mathbf{x})\, G\, \mathbf{v}_D(\mathbf{x})$ and $G \succcurlyeq 0$

One Answer when $\mathbf{K} = \{\mathbf{x} \in \mathbb{R}^n : g_j(\mathbf{x}) \geqslant 0\}$

Hybrid **SYMBOLIC/NUMERIC** methods

[M.-Allamigeon-Gaubert-Werner 14]

$$f \simeq \tilde{\sigma}_0 + \tilde{\sigma}_1 \, g_1 + \cdots + \tilde{\sigma}_m \, g_m$$

$$u = f - \tilde{\sigma}_0 + \tilde{\sigma}_1 \, g_1 + \cdots + \tilde{\sigma}_m \, g_m$$

One Answer when $\mathbf{K} = \{\mathbf{x} \in \mathbb{R}^n : g_j(\mathbf{x}) \geqslant 0\}$

Hybrid **SYMBOLIC**/**NUMERIC** methods

[M.-Allamigeon-Gaubert-Werner 14]

$f \simeq \tilde{\sigma}_0 + \tilde{\sigma}_1\, g_1 + \cdots + \tilde{\sigma}_m\, g_m$

$u = f - \tilde{\sigma}_0 + \tilde{\sigma}_1\, g_1 + \cdots + \tilde{\sigma}_m\, g_m$

$\boxed{\simeq \quad \rightarrow \quad =}$

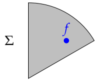
$\forall \mathbf{x} \in [0,1]^n, u(\mathbf{x}) \leqslant -\varepsilon$

$\min_{\mathbf{K}} f \geqslant \varepsilon$ when $\varepsilon \to 0$

Compact $\mathbf{K} \subseteq [0,1]^n$

From Approximate to Exact Solutions

Win Two-Player Game

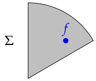


sum of squares of f?

$\simeq$ Output!

From Approximate to Exact Solutions

Win Two-Player Game



Hybrid Symbolic/Numeric Algorithms

sum of squares of $f + \varepsilon$?

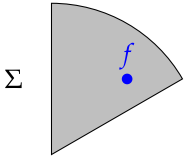
Error Compensation

$\simeq \quad \rightarrow \quad =$

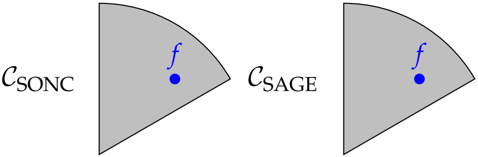
$\simeq$ Output!

From Approximate to Exact Solutions

Exact SOS



Exact SONC/SAGE








Software: RealCertify, POEM & SONCSOCP

Exact optimization via SOS: RealCertify

Maple & arbitrary precision SDP solver SDPA-GMP [Nakata 10]

univsos	$n = 1$
multivsos	$n > 1$

Exact optimization via SONC/SAGE: POEM SONCSOCP

Python (SymPy) & geometric programming/relative entropy ECOS [Domahidi-Chu-Boyd 13]

Julia & second-order programming Mosek [Andersen-Andersen 00]

Univariate SOS

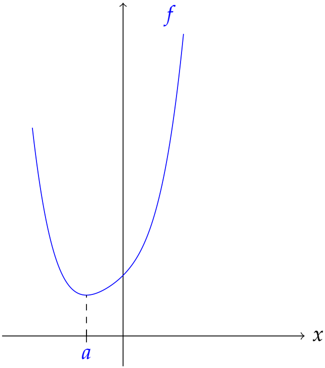
Multivariate SOS

SONC/SAGE

Benchmarks

univsos1: outline [Schweighofer 99]

$f \in \mathbb{Q}[x]$ and $f > 0$
Minimizer a may not be in $\mathbb{Q}$ ...



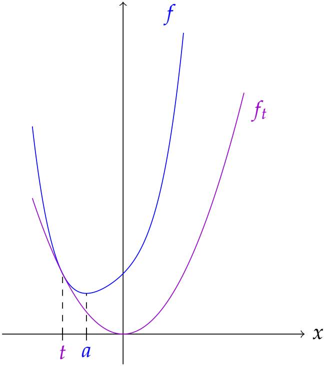
$f = 1 + x + x^2 + x^3 + x^4$
$a = \frac{5}{4(135+60\sqrt{6})^{1/3}} - \frac{4(135+60\sqrt{6})^{1/3}}{12} - \frac{1}{4}$

`univsos1`: outline [Schweighofer 99]

$f \in \mathbb{Q}[x]$ and $f > 0$
Minimizer a may not be in $\mathbb{Q}$ . . .

Find $f_t \in \mathbb{Q}[x]$ s.t. :

- $\deg f_t \leqslant 2$
- $f_t \geqslant 0$
- $f \geqslant f_t$
- $f - f_t$ has a root $t \in \mathbb{Q}$



$f = 1 + x + x^2 + x^3 + x^4$

$a = \frac{5}{4(135+60\sqrt{6})^{1/3}} - \frac{4(135+60\sqrt{6})^{1/3}}{12} - \frac{1}{4}$

$f_t = x^2$

$t = -1$

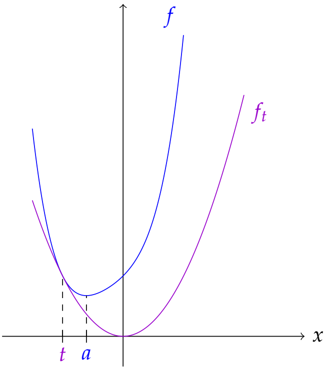
univsos1: outline [Schweighofer 99]

$f \in \mathbb{Q}[x]$ and $f > 0$
Minimizer a may not be in $\mathbb{Q}$ . . .

💡 Square-free decomposition:

$$f - f_t = gh^2$$

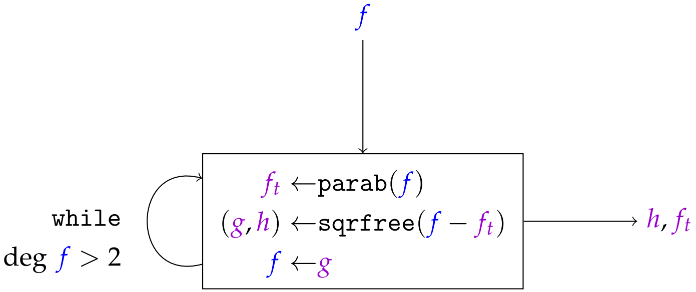
- $\deg g \leqslant \deg f - 2$
- $g > 0$
- Do it again on g



$f = 1 + x + x^2 + x^3 + x^4$
$f_t = x^2$
$f - f_t = (x^2 + 2x + 1)(x + 1)^2$

`univsos1`: algorithm [Schweighofer 99]

- **Input**: $f \geqslant 0 \in \mathbb{Q}[x]$ of degree $d \geqslant 2$
- **Output**: SOS decomposition with coefficients in $\mathbb{Q}$



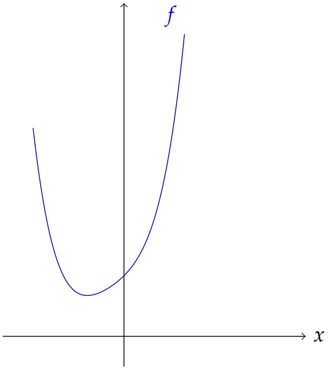
`univsos1`: output bitsize

Theorem [M.-Safey El Din-Schweighofer 17]

Let $0 < f \in \mathbb{Q}[x]$ with bitsize τ, $\deg f = d$.
The output bitsize of `univsos1` on f is $\mathcal{O}\left(\left(\frac{d}{2}\right)^{\frac{3d}{2}}\tau\right)$.

univsos2 [Chevillard et. al 11]

$f \in \mathbb{Q}[x]$, deg $f = d = 2k$, $f > 0$



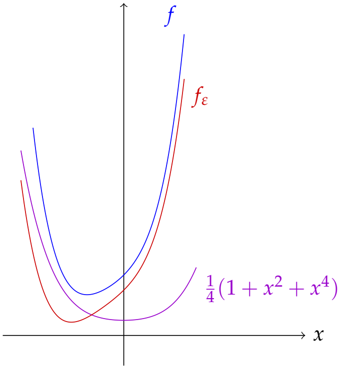
$f = 1 + x + x^2 + x^3 + x^4$

univsos2 [Chevillard et. al 11]

$f \in \mathbb{Q}[x]$, $\deg f = d = 2k$, $f > 0$

PERTURB: find $\varepsilon \in \mathbb{Q}$ s.t.

$$f_\varepsilon := f - \varepsilon \sum_{i=0}^{k} x^{2i} > 0$$



$$f = 1 + x + x^2 + x^3 + x^4$$

$$\varepsilon = \frac{1}{4}$$

$$f > \frac{1}{4}(1 + x^2 + x^4)$$

`univsos2` [Chevillard et. al 11]

$f \in \mathbb{Q}[x]$, deg $f = d = 2k$, $f > 0$

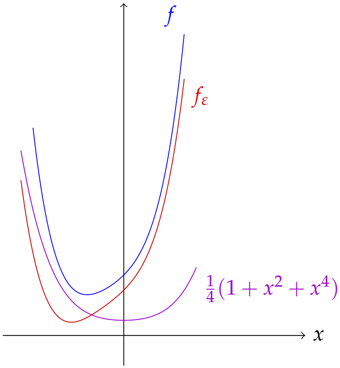
Perturb: find $\varepsilon \in \mathbb{Q}$ s.t.

$$f_\varepsilon := f - \varepsilon \sum_{i=0}^{k} x^{2i} > 0$$

Root isolation or SDP:

$$f - \varepsilon \sum_{i=0}^{k} x^{2i} = \tilde{\sigma} + u$$

Absorb: small enough u_i
$\implies \varepsilon \sum_{i=0}^{k} x^{2i} + u$ SOS



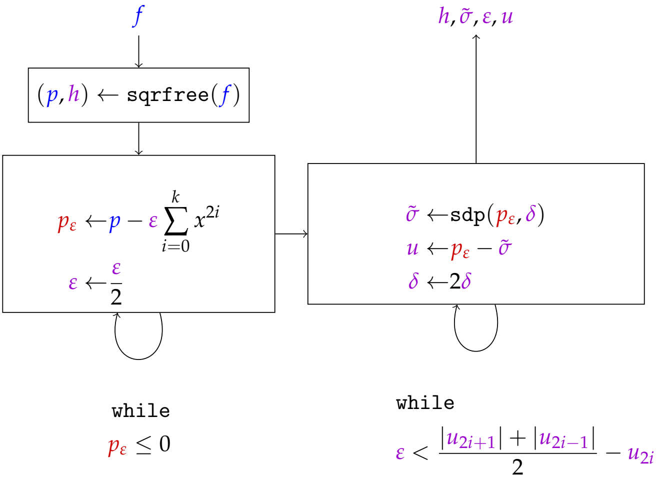
$$f = 1 + x + x^2 + x^3 + x^4$$

$$\varepsilon = \frac{1}{4}$$

$$f > \frac{1}{4}(1 + x^2 + x^4)$$

univsos2: algorithm

- **Input** $f \geqslant 0 \in \mathbb{Q}[x]$ of degree $d \geqslant 2$, $\varepsilon \in \mathbb{Q}^{>0}$, $\delta \in \mathbb{N}^{>0}$
- **Output**: SOS decomposition with coefficients in $\mathbb{Q}$



univsos2: absorb

- $x = \frac{1}{2}\left[(x+1)^2 - 1 - x^2\right]$
- $-x = \frac{1}{2}\left[(x-1)^2 - 1 - x^2\right]$

univsos2: absorb

- $x = \frac{1}{2}\left[(x+1)^2 - 1 - x^2\right]$
- $-x = \frac{1}{2}\left[(x-1)^2 - 1 - x^2\right]$

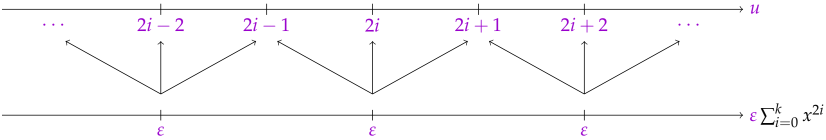
$$u_{2i+1} x^{2i+1} = \frac{|u_{2i+1}|}{2}\left[(x^{i+1} + \operatorname{sgn}(u_{2i+1}) x^i)^2 - x^{2i} - x^{2i+2}\right]$$

univsos2: absorb

$$x = \frac{1}{2}\left[(x+1)^2 - 1 - x^2\right]$$

$$-x = \frac{1}{2}\left[(x-1)^2 - 1 - x^2\right]$$

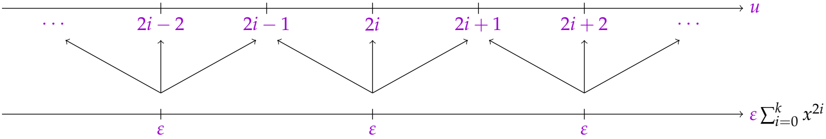
$$u_{2i+1} x^{2i+1} = \frac{|u_{2i+1}|}{2}\left[(x^{i+1} + \operatorname{sgn}(u_{2i+1}) x^i)^2 - x^{2i} - x^{2i+2}\right]$$



univsos2: absorb

$$x = \frac{1}{2}\left[(x+1)^2 - 1 - x^2\right]$$
$$-x = \frac{1}{2}\left[(x-1)^2 - 1 - x^2\right]$$

$$u_{2i+1} x^{2i+1} = \frac{|u_{2i+1}|}{2}\left[(x^{i+1} + \operatorname{sgn}(u_{2i+1}) x^i)^2 - x^{2i} - x^{2i+2}\right]$$



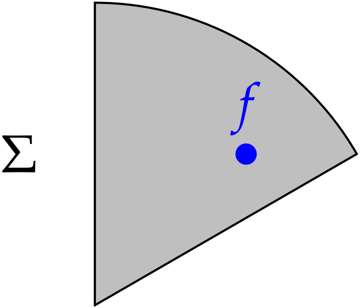
$$\varepsilon \geqslant \frac{|u_{2i+1}| + |u_{2i-1}|}{2} - u_{2i} \implies \varepsilon \sum_{i=0}^{k} x^{2i} + u \quad \text{SOS}$$

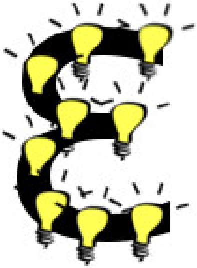
univsos2: output bitsize

Theorem [M.-Safey El Din-Schweighofer 17]

Let $0 < f \in \mathbb{Q}[x]$ with bitsize τ, $\deg f = d$.
The output bitsize of `univsos2` on f is $\mathcal{O}(d^3 + d^2\tau)$.

multivsos: perturbation





PERTURBATION idea

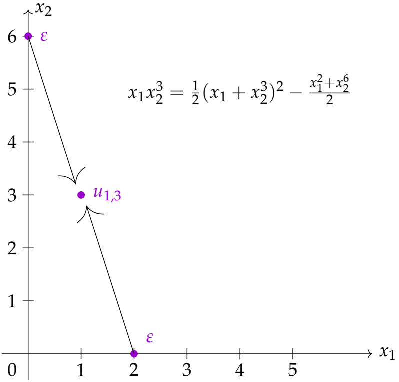
Approximate SOS Decomposition

$$f(\mathbf{x}) - \varepsilon \sum_{\alpha \in \mathcal{P}/2} \mathbf{x}^{2\alpha} = \tilde{\sigma} + u$$

multivsos: absorb

$$f(x) - \varepsilon \sum_{\alpha \in \mathcal{P}/2} x^{2\alpha} = \tilde{\sigma} + u$$

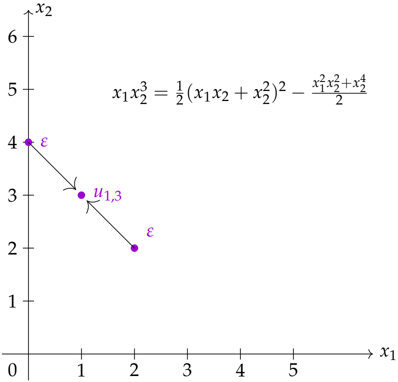
Choice of $\mathcal{P}$?



multivsos: absorb

$$f(x) - \varepsilon \sum_{\alpha \in \mathcal{P}/2} x^{2\alpha} = \tilde{\sigma} + u$$

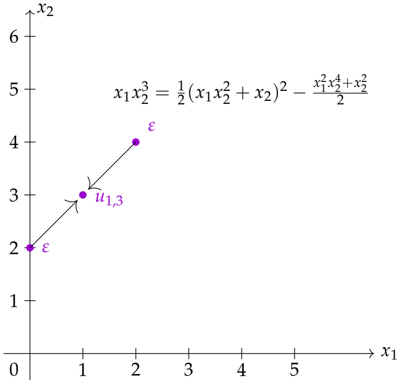
Choice of $\mathcal{P}$?



multivsos: absorb

$$f(x) - \varepsilon \sum_{\alpha \in \mathcal{P}/2} x^{2\alpha} = \tilde{\sigma} + u$$

Choice of $\mathcal{P}$?

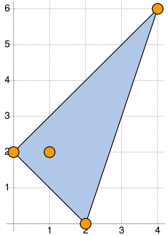


multivsos: absorb

$$f(x) - \varepsilon \sum_{\alpha \in \mathcal{P}/2} x^{2\alpha} = \tilde{\sigma} + u$$

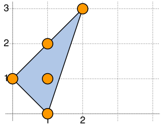
Choice of $\mathcal{P}$?

$f = 4x_1^4 x_2^6 + x_1^2 - x_1 x_2^2 + x_2^2$
spt(f) = $\{(4,6),(2,0),(1,2),(0,2)\}$



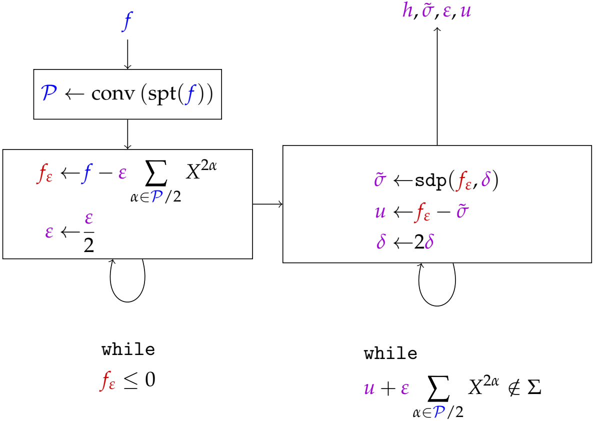
Newton Polytope $\mathcal{P} = \operatorname{conv}(\text{spt}(f))$

Squares in SOS decomposition $\subseteq \frac{\mathcal{P}}{2} \cap \mathbb{N}^n$
[Reznick 78]



multivsos: algorithm

- **Input** $f \in \mathbb{Q}[\mathbf{x}] \cap \mathring{\Sigma}[\mathbf{x}]$ of degree d, $\varepsilon \in \mathbb{Q}^{>0}$, $\delta \in \mathbb{N}^{>0}$
- **Output**: SOS decomposition with coefficients in $\mathbb{Q}$



multivsos: output bitsize

Theorem (Exact Certification Cost in $\mathring{\Sigma}$)

$f \in \mathbb{Q}[\mathbf{x}] \cap \mathring{\Sigma}[\mathbf{x}]$ with $\deg f = d = 2k$ and bit size τ

$\Longrightarrow$ `multivsos` terminates with SOS output of bit size $\boxed{\tau\, d^{d^{\mathcal{O}(n)}}}$

Algorithm `Reznicksos`

f positive definite form has **Reznick**'s representation:

$$f = \frac{\sigma}{(x_1 + \cdots + x_n)^{2D}} \quad \text{with } \sigma \in \Sigma[\mathbf{x}]$$

Algorithm `Reznicksos`

f positive definite form has **Reznick**'s representation:

$$f = \frac{\sigma}{(x_1 + \cdots + x_n)^{2D}} \quad \text{with } \sigma \in \Sigma[\mathbf{x}]$$

Theorem

$$f\,(x_1 + \cdots + x_n)^{2D} \in \Sigma[\mathbf{x}] \implies f\,(x_1 + \cdots + x_n)^{2D+2} \in \mathring{\Sigma}[\mathbf{x}]$$

Algorithm `Reznicksos`

f positive definite form has **Reznick**'s representation:

$$f = \frac{\sigma}{(x_1 + \cdots + x_n)^{2D}} \quad \text{with } \sigma \in \Sigma[\mathbf{x}]$$

Theorem

$$f\,(x_1 + \cdots + x_n)^{2D} \in \Sigma[\mathbf{x}] \implies f\,(x_1 + \cdots + x_n)^{2D+2} \in \mathring{\Sigma}[\mathbf{x}]$$

Apply Algorithm `multivsos` on $f\,(x_1 + \cdots + x_n)^{2D+2}$

Algorithm `Reznicksos`

f positive definite form has **Reznick**'s representation:

$$f = \frac{\sigma}{(x_1 + \cdots + x_n)^{2D}} \quad \text{with } \sigma \in \Sigma[\mathbf{x}]$$

Theorem

$$f\,(x_1 + \cdots + x_n)^{2D} \in \Sigma[\mathbf{x}] \implies f\,(x_1 + \cdots + x_n)^{2D+2} \in \mathring{\Sigma}[\mathbf{x}]$$

Apply Algorithm `multivsos` on $f\,(x_1 + \cdots + x_n)^{2D+2}$

Theorem (Exact Certification Cost of Reznick's representations)

$f \in \mathbb{Q}[\mathbf{x}]$ positive definite form with $\deg f = d$ and bit size τ

$$\implies D \leqslant 2^{2^{\mathcal{O}\left(\tau \cdot (4d+6)^{3n+3}\right)}}$$

Algorithm `Putinarsos`

Assumption: $\exists i$ s.t. $g_i = 1 - \|\mathbf{x}\|_2^2$

$f > 0$ on $\mathbf{K} := \{\mathbf{x} : g_j(\mathbf{x}) \geqslant 0\}$ has **Putinar**'s representation:

$$f = \sigma_0 + \sum_j \sigma_j \, g_j \quad \text{with } \sigma_j \in \Sigma[\mathbf{x}] \, , \deg \sigma_j \leqslant 2D$$

Algorithm `Putinarsos`

Assumption: $\exists i$ s.t. $g_i = 1 - \|\mathbf{x}\|_2^2$

$f > 0$ on $\mathbf{K} := \{\mathbf{x} : g_j(\mathbf{x}) \geqslant 0\}$ has **Putinar**'s representation:

$$f = \sigma_0 + \sum_j \sigma_j \, g_j \quad \text{with } \sigma_j \in \Sigma[\mathbf{x}] \, , \deg \sigma_j \leqslant 2D$$

Theorem [M.-Safey El Din 18]

$$f = \mathring{\sigma}_0 + \sum_j \mathring{\sigma}_j \, g_j$$

with $\mathring{\sigma}_j \in \mathring{\Sigma}[\mathbf{x}] \, , \deg \mathring{\sigma}_j \leqslant 2D$

Algorithm `Putinarsos`

Assumption: $\exists i$ s.t. $g_i = 1 - \|\mathbf{x}\|_2^2$

$f > 0$ on $\mathbf{K} := \{\mathbf{x} : g_j(\mathbf{x}) \geqslant 0\}$ has **Putinar**'s representation:

$$f = \sigma_0 + \sum_j \sigma_j \, g_j \quad \text{with } \sigma_j \in \Sigma[\mathbf{x}] \, , \deg \sigma_j \leqslant 2D$$

Theorem [M.-Safey El Din 18]

$$f = \mathring{\sigma}_0 + \sum_j \mathring{\sigma}_j \, g_j$$

$$\text{with } \mathring{\sigma}_j \in \mathring{\Sigma}[\mathbf{x}] \, , \deg \mathring{\sigma}_j \leqslant 2D$$

ABSORB as in Algorithm `multivsos`: $u = f_\varepsilon - \tilde{\sigma}_0 - \sum_j \tilde{\sigma}_j \, g_j$

Algorithm `Putinarsos`

Assumption: $\exists i$ s.t. $g_i = 1 - \|\mathbf{x}\|_2^2$

$f > 0$ on $\mathbf{K} := \{\mathbf{x} : g_j(\mathbf{x}) \geqslant 0\}$ has **Putinar**'s representation:

$$f = \sigma_0 + \sum_j \sigma_j g_j \quad \text{with } \sigma_j \in \Sigma[\mathbf{x}], \deg \sigma_j \leqslant 2D$$

Theorem [M.-Safey El Din 18]

$$f = \mathring{\sigma}_0 + \sum_j \mathring{\sigma}_j g_j$$

$$\text{with } \mathring{\sigma}_j \in \mathring{\Sigma}[\mathbf{x}], \deg \mathring{\sigma}_j \leqslant 2D$$

ABSORB as in Algorithm `multivsos`: $u = f_\varepsilon - \tilde{\sigma}_0 - \sum_j \tilde{\sigma}_j g_j$

OUTPUT BIT SIZE = $\boxed{D^{D^{\mathcal{O}(n)}} \text{ with } \log D = \mathcal{O}\left(2^{\tau d^{n C_{\mathbf{K}}}}\right)}$

gradsos: SOS modulo the gradient ideal

Theorem [M.-Safey El Din-Vu 21]

Let $f \in \mathbb{Q}[\mathbf{x}]$. Assume that

1. The infimum $f^\star$ is attained
2. The gradient ideal $\mathcal{I}_{\text{grad}}(f)$ is zero-dimensional radical

$\implies$ f is nonnegative over $\mathbb{R}^n$ $\Leftrightarrow$ f is SOS modulo $\mathcal{I}_{\text{grad}}(f)$,

i.e. there exists $q_j, \phi_i \in \mathbb{Q}(\mathbf{x})$, $g_i \in \mathcal{I}_{\text{grad}}(f)$ s.t. $f = \sum_{j=1}^{s} q_j^2 + \sum_{i=1}^{m} \phi_i g_i$.

gradsos: SOS modulo the gradient ideal

Theorem [M.-Safey El Din-Vu 21]

Let $f \in \mathbb{Q}[\mathbf{x}]$. Assume that

1. The infimum $f^\star$ is attained
2. The gradient ideal $\mathcal{I}_{\text{grad}}(f)$ is zero-dimensional radical

$\Longrightarrow$ f is nonnegative over $\mathbb{R}^n$ $\Leftrightarrow$ f is SOS modulo $\mathcal{I}_{\text{grad}}(f)$,

i.e. there exists $q_j, \phi_i \in \mathbb{Q}(\mathbf{x})$, $g_i \in \mathcal{I}_{\text{grad}}(f)$ s.t. $f = \sum_{j=1}^{s} q_j^2 + \sum_{i=1}^{m} \phi_i g_i$.

Idea of the proof: reduce the number of variables from n to 1.

gradsos: SOS modulo the gradient ideal

Theorem [M.-Safey El Din-Vu 21]

Let $f \in \mathbb{Q}[\mathbf{x}]$. Assume that

1. The infimum $f^\star$ is attained
2. The gradient ideal $\mathcal{I}_{\text{grad}}(f)$ is zero-dimensional radical

$\implies$ f is nonnegative over $\mathbb{R}^n$ $\Leftrightarrow$ f is SOS modulo $\mathcal{I}_{\text{grad}}(f)$,

i.e. there exists $q_j, \phi_i \in \mathbb{Q}(\mathbf{x})$, $g_i \in \mathcal{I}_{\text{grad}}(f)$ s.t. $f = \sum_{j=1}^{s} q_j{}^2 + \sum_{i=1}^{m} \phi_i g_i$.

Idea of the proof: reduce the number of variables from n to 1.

Robinson and Scheiderer's polynomials satisfy both conditions

$$f_R = x_1^6 + x_2^6 - x_1^4 x_2^2 + 3x_1^2 x_2^2 - x_1^2 x_2^4 - x_1^4 - x_2^4 - x_1^2 - x_2^2 + 1$$
$$f_S = x_1^4 + x_1 x_2^3 + x_2^4 + 3x_1^2 x_2 + 4x_1 x_2^2 + 2x_1^2 - x_1 - x_2 + 1$$

$\Rightarrow$ They are SOS modulo their gradient ideals over $\mathbb{Q}[\mathbf{x}]$

gradsos: algorithm and bitsize

1: Find a 0-dim rational parametrization of $V_{\text{grad}}(f)$:
$\{w, \frac{\kappa_1}{w'}, \ldots, \frac{\kappa_n}{w'}\}$

gradsos: algorithm and bitsize

1: Find a 0-dim rational parametrization of $V_{\text{grad}}(f)$:
$\{w, \frac{\kappa_1}{w'}, \ldots, \frac{\kappa_n}{w'}\}$

2: $h := (w')^d f(x_1, \frac{\kappa_2}{w'}, \ldots, \frac{\kappa_n}{w'})$

gradsos: algorithm and bitsize

1: Find a 0-dim rational parametrization of $V_{\text{grad}}(f)$: $\{w, \frac{\kappa_1}{w'}, \ldots, \frac{\kappa_n}{w'}\}$

2: $h := (w')^d f(x_1, \frac{\kappa_2}{w'}, \ldots, \frac{\kappa_n}{w'})$

3: Find $\phi_1, \ldots, \phi_n$ satisfying

$$(w')^d f - h = \sum_{i=1}^n \phi_i (x_i - \frac{\kappa_i}{w'})$$

by using divisions of univariate polynomials

gradsos: algorithm and bitsize

1: Find a 0-dim rational parametrization of $V_{\text{grad}}(f)$:
$\{w, \frac{\kappa_1}{w'}, \dots, \frac{\kappa_n}{w'}\}$

2: $h := (w')^d f(x_1, \frac{\kappa_2}{w'}, \dots, \frac{\kappa_n}{w'})$

3: Find $\phi_1, \dots, \phi_n$ satisfying

$$(w')^d f - h = \sum_{i=1}^n \phi_i (x_i - \frac{\kappa_i}{w'})$$

by using divisions of univariate polynomials

4: Find an SOS decomposition of h by using `univsos1` or `univsos2`

gradsos: algorithm and bitsize

1: Find a 0-dim rational parametrization of $V_{\text{grad}}(f)$: $\{w, \frac{\kappa_1}{w'}, \ldots, \frac{\kappa_n}{w'}\}$

2: $h := (w')^d f(x_1, \frac{\kappa_2}{w'}, \ldots, \frac{\kappa_n}{w'})$

3: Find $\phi_1, \ldots, \phi_n$ satisfying

$$(w')^d f - h = \sum_{i=1}^n \phi_i (x_i - \frac{\kappa_i}{w'})$$

by using divisions of univariate polynomials

4: Find an SOS decomposition of h by using `univsos1` or `univsos2`

Theorem

With $f \in \mathbb{Q}[\mathbf{x}]$ with $\deg f = d$ and bitsize τ

`gradsos` terminates with output bitsize $\boxed{\widetilde{\mathcal{O}}\left((\tau + n + d)d^{3n+2}\right)}$

Algorithm optsonc: numerical steps

SONC (SUMS OF NONNEGATIVE CIRCUITS)

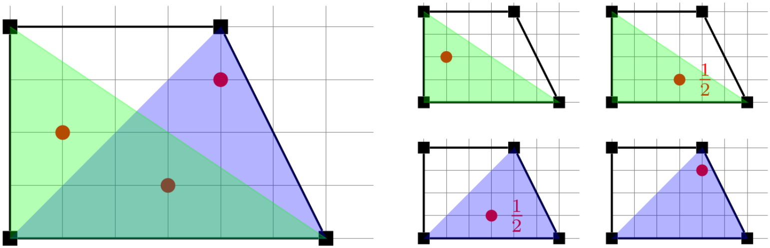
- **Input** $f = \sum_{\boldsymbol{\alpha}} b_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}}$ of degree d, $\hat{\delta} \in \mathbb{Q}^{>0}$, $\tilde{\delta} \in \mathbb{Q}^{>0}$
  Monomial squares = $\Lambda(f)$ Complement = $\Gamma(f)$

Algorithm optsonc: numerical steps

SONC (SUMS OF NONNEGATIVE CIRCUITS)

- **Input** $f = \sum_{\alpha} b_{\alpha} \mathbf{x}^{\alpha}$ of degree d, $\hat{\delta} \in \mathbb{Q}^{>0}$, $\tilde{\delta} \in \mathbb{Q}^{>0}$
  Monomial squares = $\Lambda(f)$ Complement = $\Gamma(f)$

1. **Cover** each $\beta \in \Gamma(f)$ to get nonnegative circuit f_{β}
   $\implies \lambda^{\beta} \geqslant 0$ with $\sum_{\alpha \in \Lambda(f)} \lambda_{\alpha}^{\beta} \cdot \alpha = \beta$



Algorithm `optsonc`: numerical steps

- **Input** $f = \sum_{\boldsymbol{\alpha}} b_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}}$, $\hat{\delta}$, $\tilde{\delta}$

2. **Numerical** resolution of GEOMETRIC PROGRAM

$$
\begin{aligned}
f_{\text{SONC}} = \min_{\boldsymbol{G} > \mathbf{0}} \quad & \sum_{\boldsymbol{\beta} \in \Gamma(f)} G_{\boldsymbol{\beta},\mathbf{0}} \\
\text{s.t.} \quad & \sum_{\boldsymbol{\beta} \in \Gamma(f)} G_{\boldsymbol{\beta},\boldsymbol{\alpha}} \leqslant b_{\boldsymbol{\alpha}} \,, \quad \boldsymbol{\alpha} \in \Lambda(f) \,, \boldsymbol{\alpha} \neq \mathbf{0} \\
& \prod_{\boldsymbol{\alpha} \in \text{Cov}^{\boldsymbol{\beta}}} \left(\frac{G_{\boldsymbol{\beta},\boldsymbol{\alpha}}}{\lambda_{\boldsymbol{\alpha}}^{\boldsymbol{\beta}}} \right)^{\lambda_{\boldsymbol{\alpha}}^{\boldsymbol{\beta}}} = -b_{\boldsymbol{\beta}} \,, \quad \boldsymbol{\beta} \in \Gamma(f)
\end{aligned}
$$

Algorithm `optsonc`: symbolic steps

- **Input** $f = \sum_{\alpha} b_{\alpha} \mathbf{x}^{\alpha}$, $\hat{\delta}$, $\tilde{\delta}$

GEOMETRIC PROGRAM provides "IN THEORY"

$f_{\beta} = \sum_{\alpha} G_{\beta,\alpha} \cdot \mathbf{x}^{\alpha} + b_{\beta} \mathbf{x}^{\beta}$, $f + \sum_{\beta} G_{\beta,\mathbf{0}} - b_{\mathbf{0}} = \sum_{\beta} f_{\beta} \geqslant 0$

Algorithm `optsonc`: symbolic steps

- **Input** $f = \sum_{\boldsymbol{\alpha}} b_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}}$, $\hat{\delta}$, $\tilde{\delta}$

GEOMETRIC PROGRAM provides "IN THEORY"

$f_{\boldsymbol{\beta}} = \sum_{\boldsymbol{\alpha}} G_{\boldsymbol{\beta},\boldsymbol{\alpha}} \cdot \mathbf{x}^{\boldsymbol{\alpha}} + b_{\boldsymbol{\beta}} \mathbf{x}^{\boldsymbol{\beta}}$, $f + \sum_{\boldsymbol{\beta}} G_{\boldsymbol{\beta},\mathbf{0}} - b_{\mathbf{0}} = \sum_{\boldsymbol{\beta}} f_{\boldsymbol{\beta}} \geqslant 0$

Tolerance $\tilde{\delta} \implies$ "IN PRACTICE" $\tilde{G}$ violates the constraints

Algorithm `optsonc`: symbolic steps

- **Input** $f = \sum_{\alpha} b_{\alpha} \mathbf{x}^{\alpha}$, $\hat{\delta}$, $\tilde{\delta}$

GEOMETRIC PROGRAM provides "IN THEORY"

$f_{\beta} = \sum_{\alpha} G_{\beta,\alpha} \cdot \mathbf{x}^{\alpha} + b_{\beta} \mathbf{x}^{\beta}$, $f + \sum_{\beta} G_{\beta,\mathbf{0}} - b_{\mathbf{0}} = \sum_{\beta} f_{\beta} \geqslant 0$

Tolerance $\tilde{\delta} \implies$ "IN PRACTICE" $\tilde{G}$ violates the constraints

3. **Rounding step** $\hat{G} \leftarrow \texttt{round}\left(\tilde{G}, \hat{\delta}\right)$

Algorithm `optsonc`: symbolic steps

- **Input** $f = \sum_{\alpha} b_{\alpha} \mathbf{x}^{\alpha}$, $\hat{\delta}$, $\tilde{\delta}$

GEOMETRIC PROGRAM provides "IN THEORY"

$f_{\beta} = \sum_{\alpha} G_{\beta,\alpha} \cdot \mathbf{x}^{\alpha} + b_{\beta} \mathbf{x}^{\beta}$, $f + \sum_{\beta} G_{\beta,\mathbf{0}} - b_{\mathbf{0}} = \sum_{\beta} f_{\beta} \geqslant 0$

Tolerance $\tilde{\delta} \implies$ "IN PRACTICE" $\tilde{G}$ violates the constraints

3. **Rounding step** $\hat{G} \leftarrow \texttt{round}\left(\tilde{G}, \hat{\delta}\right)$
4. **Projection step**

$$G_{\beta,\alpha} \leftarrow b_{\alpha} \cdot \hat{G}_{\beta,\alpha} / \sum_{\beta'} \hat{G}_{\beta',\alpha}$$

$$\tilde{G}_{\beta,\mathbf{0}} \leftarrow \lambda_{\mathbf{0}}^{\beta} \left(-b_{\beta} \cdot \prod_{\alpha} \left(\frac{\lambda_{\alpha}^{\beta}}{G_{\beta,\alpha}} \right)^{\lambda_{\alpha}^{\beta}} \right)^{\frac{1}{\lambda_{\mathbf{0}}^{\beta}}}$$

$$\hat{G}_{\beta,\mathbf{0}} \leftarrow \texttt{round}\uparrow \left(\tilde{G}_{\beta,\mathbf{0}}, \hat{\delta} \right)$$

Algorithm `optsonc`: symbolic steps

- **Input** $f = \sum_{\alpha} b_{\alpha} \mathbf{x}^{\alpha}$, $\hat{\delta}$, $\tilde{\delta}$

GEOMETRIC PROGRAM provides "IN THEORY"

$f_{\beta} = \sum_{\alpha} G_{\beta,\alpha} \cdot \mathbf{x}^{\alpha} + b_{\beta} \mathbf{x}^{\beta}$, $f + \sum_{\beta} G_{\beta,\mathbf{0}} - b_{\mathbf{0}} = \sum_{\beta} f_{\beta} \geqslant 0$

Tolerance $\tilde{\delta} \implies$ "IN PRACTICE" $\tilde{G}$ violates the constraints

3. **Rounding step** $\hat{G} \leftarrow \texttt{round}\left(\tilde{G}, \hat{\delta}\right)$
4. **Projection step**

$$G_{\beta,\alpha} \leftarrow b_{\alpha} \cdot \hat{G}_{\beta,\alpha} / \sum_{\beta'} \hat{G}_{\beta',\alpha}$$

$$\tilde{G}_{\beta,\mathbf{0}} \leftarrow \lambda_{\mathbf{0}}^{\beta} \left(-b_{\beta} \cdot \prod_{\alpha} \left(\frac{\lambda_{\alpha}^{\beta}}{G_{\beta,\alpha}} \right)^{\lambda_{\alpha}^{\beta}} \right)^{\frac{1}{\lambda_{\mathbf{0}}^{\beta}}}$$

$$\hat{G}_{\beta,\mathbf{0}} \leftarrow \texttt{round}\uparrow \left(\tilde{G}_{\beta,\mathbf{0}}, \hat{\delta} \right)$$

$$\boxed{f \geqslant b_{\mathbf{0}} - \sum_{\beta} \hat{G}_{\beta,\mathbf{0}}}$$

Algorithm `optsonc` with SOCP

Averages of distinct rational points in M

$$A(M) := \{\frac{1}{2}(\mathbf{v}+\mathbf{w}) \mid \mathbf{v} \neq \mathbf{w}, \mathbf{v}, \mathbf{w} \in M\}$$

M is an $\mathcal{A}$-rational mediated set if $\mathcal{A} \subseteq M \subseteq A(M) \cup \mathcal{A}$

Algorithm optsonc with SOCP

Averages of distinct rational points in M

$$A(M) := \{\frac{1}{2}(\mathbf{v} + \mathbf{w}) \mid \mathbf{v} \neq \mathbf{w}, \mathbf{v}, \mathbf{w} \in M\}$$

M is an $\mathcal{A}$-rational mediated set if $\mathcal{A} \subseteq M \subseteq A(M) \cup \mathcal{A}$

Theorem [Wang-Magron 20] for a circuit f

There exists a $\Lambda(f)$-rational mediated set M containing $\boldsymbol{\beta}$

Algorithm optsonc with SOCP

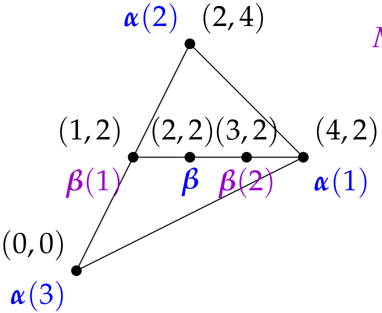
Averages of distinct rational points in M

$$A(M) := \{\frac{1}{2}(\mathbf{v}+\mathbf{w}) \mid \mathbf{v} \neq \mathbf{w}, \mathbf{v}, \mathbf{w} \in M\}$$

M is an $\mathcal{A}$-rational mediated set if $\mathcal{A} \subseteq M \subseteq A(M) \cup \mathcal{A}$

Theorem [Wang-Magron 20] for a circuit f

There exists a $\Lambda(f)$-rational mediated set M containing $\boldsymbol{\beta}$



$$M = \{\boldsymbol{\alpha}(1), \boldsymbol{\alpha}(2), \boldsymbol{\alpha}(3), \boldsymbol{\beta}, \boldsymbol{\beta}(1), \boldsymbol{\beta}(2)\}$$

$$\mathbf{u} \in M \backslash \Lambda(f) \implies \mathbf{u} = (\mathbf{v}+\mathbf{w})/2$$

Algorithm `optsonc` with SOCP

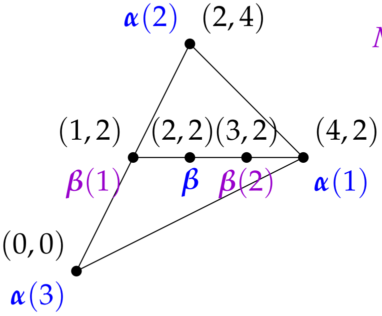
Averages of distinct rational points in M

$$A(M) := \{\frac{1}{2}(\mathbf{v}+\mathbf{w}) \mid \mathbf{v} \neq \mathbf{w}, \mathbf{v}, \mathbf{w} \in M\}$$

M is an $\mathcal{A}$-rational mediated set if $\mathcal{A} \subseteq M \subseteq A(M) \cup \mathcal{A}$

Theorem [Wang-Magron 20] for a circuit f

There exists a $\Lambda(f)$-rational mediated set M containing $\boldsymbol{\beta}$



$$M = \{\boldsymbol{\alpha}(1), \boldsymbol{\alpha}(2), \boldsymbol{\alpha}(3), \boldsymbol{\beta}, \boldsymbol{\beta}(1), \boldsymbol{\beta}(2)\}$$

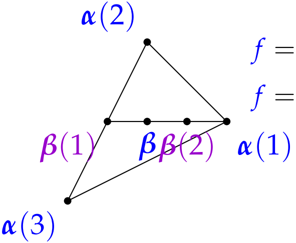
$$\mathbf{u} \in M \backslash \Lambda(f) \implies \mathbf{u} = (\mathbf{v}+\mathbf{w})/2$$

There exist a_i, b_i, c_i such that

$$f(\mathbf{x}) = \sum_i a_i \mathbf{x}^{\mathbf{v}_i} + b_i \mathbf{x}^{\mathbf{w}_i} - 2c_i \mathbf{x}^{\mathbf{u}_i}$$

$$a_i b_i \geqslant c_i^2, \quad a_i, b_i \geqslant 0 \quad \text{(SOCP)}$$

Circuits and sums of binomial squares

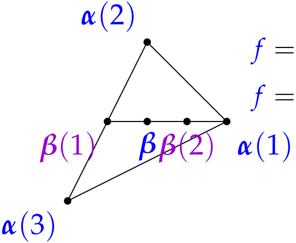


$$f = \sum_i (\sqrt{a_i}\mathbf{x}^{\frac{\mathbf{v}_i}{2}} - \sqrt{b_i}\mathbf{x}^{\frac{\mathbf{w}_i}{2}})^2$$

$$f = (1 - x_1 x_2^2)^2 + 2(x_1^{\frac{1}{2}} x_2 - x_1^{\frac{3}{2}} x_2)^2 + (x_1 x_2 - x_1^2 x_2)^2$$

$\implies$ sum of 3 binomial squares

Circuits and sums of binomial squares



$$f = \sum_i (\sqrt{a_i}\mathbf{x}^{\frac{\mathbf{v}_i}{2}} - \sqrt{b_i}\mathbf{x}^{\frac{\mathbf{w}_i}{2}})^2$$

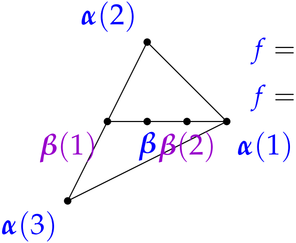
$$f = (1 - x_1 x_2^2)^2 + 2(x_1^{\frac{1}{2}} x_2 - x_1^{\frac{3}{2}} x_2)^2 + (x_1 x_2 - x_1^2 x_2)^2$$

$\implies$ sum of 3 binomial squares

“Arbitrary support” $f = \sum_{\boldsymbol{\alpha} \in \Lambda(f)} c_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}} - \sum_{\boldsymbol{\beta} \in \Gamma(f)} d_{\boldsymbol{\beta}} \mathbf{x}^{\boldsymbol{\beta}}$

Monomial squares = $\Lambda(f)$ Complement = $\Gamma(f)$

Circuits and sums of binomial squares



$$f = \sum_i (\sqrt{a_i}\mathbf{x}^{\frac{\mathbf{v}_i}{2}} - \sqrt{b_i}\mathbf{x}^{\frac{\mathbf{w}_i}{2}})^2$$

$$f = (1 - x_1 x_2^2)^2 + 2(x_1^{\frac{1}{2}} x_2 - x_1^{\frac{3}{2}} x_2)^2 + (x_1 x_2 - x_1^2 x_2)^2$$

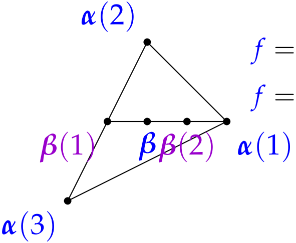
$\implies$ sum of 3 binomial squares

“Arbitrary support” $f = \sum_{\boldsymbol{\alpha} \in \Lambda(f)} c_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}} - \sum_{\boldsymbol{\beta} \in \Gamma(f)} d_{\boldsymbol{\beta}} \mathbf{x}^{\boldsymbol{\beta}}$

Monomial squares = $\Lambda(f)$ Complement = $\Gamma(f)$

Simplex **Cover** of each $\boldsymbol{\beta} \implies \mathcal{A}$ and mediated set $M_{\mathcal{A}\boldsymbol{\beta}}$

Circuits and sums of binomial squares



$f = \sum_i (\sqrt{a_i}\mathbf{x}^{\frac{\mathbf{v}_i}{2}} - \sqrt{b_i}\mathbf{x}^{\frac{\mathbf{w}_i}{2}})^2$

$f = (1 - x_1 x_2^2)^2 + 2(x_1^{\frac{1}{2}} x_2 - x_1^{\frac{3}{2}} x_2)^2 + (x_1 x_2 - x_1^2 x_2)^2$

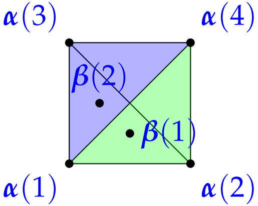
$\implies$ sum of 3 binomial squares

"Arbitrary support" $f = \sum_{\alpha \in \Lambda(f)} c_\alpha \mathbf{x}^\alpha - \sum_{\beta \in \Gamma(f)} d_\beta \mathbf{x}^\beta$

Monomial squares = $\Lambda(f)$ Complement = $\Gamma(f)$

Simplex **Cover** of each $\beta \implies \mathcal{A}$ and mediated set $M_{\mathcal{A}\beta}$

$f = 50x_1^4 x_2^4 + x_1^4 + 3x_2^4 + 800 - 100x_1 x_2^2 - 100x_1^2 x_2$



Algorithm `optsage`: numerical steps

SAGE (SUMS OF ARITHMETIC-GEOMETRIC-MEAN-EXPONENTIALS)

- **Input** $f = \sum_i b_i \exp[\mathbf{x} \cdot \boldsymbol{\alpha}(i)]$, $\hat{\delta}$, $\tilde{\delta}$

Algorithm optsage: numerical steps

SAGE (SUMS OF ARITHMETIC-GEOMETRIC-MEAN-EXPONENTIALS)

- **Input** $f = \sum_i b_i \exp[\mathbf{x} \cdot \boldsymbol{\alpha}(i)]$, $\hat{\delta}$, $\tilde{\delta}$

$f \in \mathcal{C}_{\mathsf{SAGE}} \Leftrightarrow \exists \boldsymbol{\nu}^{(j)}, \boldsymbol{c}^{(j)}$ such that

$$\sum_j \boldsymbol{c}^{(j)} = \boldsymbol{b}\,, \quad \sum_i \boldsymbol{\alpha}(i) \boldsymbol{\nu}_i^{(j)} = \mathbf{0}\,, \quad -\mathbf{1} \cdot \boldsymbol{\nu}_{\backslash j}^{(j)} = \nu_j^{(j)}$$

$$\boldsymbol{c}_{\backslash j}^{(j)}, \boldsymbol{\nu}_{\backslash j}^{(j)} \geqslant \mathbf{0}\,, \quad D\left(\boldsymbol{\nu}_{\backslash j}^{(j)}, e\boldsymbol{c}_{\backslash j}^{(j)}\right) \leqslant c_j^{(j)}$$

Algorithm `optsage`: numerical steps

SAGE (SUMS OF ARITHMETIC-GEOMETRIC-MEAN-EXPONENTIALS)

- **Input** $f = \sum_i b_i \exp[\mathbf{x} \cdot \boldsymbol{\alpha}(i)]$, $\hat{\delta}$, $\tilde{\delta}$

$f \in \mathcal{C}_{\text{SAGE}} \Leftrightarrow \exists \boldsymbol{\nu}^{(j)}, \boldsymbol{c}^{(j)}$ such that

$$\sum_j \boldsymbol{c}^{(j)} = \boldsymbol{b}\,, \quad \sum_i \boldsymbol{\alpha}(i) \boldsymbol{\nu}_i^{(j)} = \mathbf{0}\,, \quad -\mathbf{1} \cdot \boldsymbol{\nu}_{\backslash j}^{(j)} = \nu_j^{(j)}$$

$$\boldsymbol{c}_{\backslash j}^{(j)}, \boldsymbol{\nu}_{\backslash j}^{(j)} \geqslant \mathbf{0}\,, \quad D\left(\boldsymbol{\nu}_{\backslash j}^{(j)}, e\boldsymbol{c}_{\backslash j}^{(j)}\right) \leqslant c_j^{(j)}$$

1. **Numerical** resolution of RELATIVE ENTROPY PROGRAM

Precision $\tilde{\delta} \implies$ “IN PRACTICE” $\tilde{\boldsymbol{\nu}}, \tilde{\boldsymbol{c}}$ violate the constraints

Algorithm `optsage`: symbolic steps

- **Input** $f = \sum_i b_i \exp[\mathbf{x} \cdot \boldsymbol{\alpha}(i)]$, $\hat{\delta}$, $\tilde{\delta}$

Algorithm `optsage`: symbolic steps

- **Input** $f = \sum_i b_i \exp[\mathbf{x} \cdot \boldsymbol{\alpha}(i)]$, $\hat{\delta}$, $\tilde{\delta}$

Build the matrix Q with columns $(\boldsymbol{\alpha}(i), 1)$

Algorithm optsage: symbolic steps

- **Input** $f = \sum_i b_i \exp[\mathbf{x} \cdot \boldsymbol{\alpha}(i)]$, $\hat{\delta}$, $\tilde{\delta}$

Build the matrix Q with columns $(\boldsymbol{\alpha}(i), 1)$

2. **Rounding step** $\hat{\boldsymbol{v}} \leftarrow \texttt{round}\left(\tilde{\boldsymbol{v}}, \hat{\delta}\right)$, $\hat{\boldsymbol{c}} \leftarrow \texttt{round}\left(\tilde{\boldsymbol{c}}, \hat{\delta}\right)$

Algorithm `optsage`: symbolic steps

- **Input** $f = \sum_i b_i \exp[\mathbf{x} \cdot \boldsymbol{\alpha}(i)]$, $\hat{\delta}$, $\tilde{\delta}$

Build the matrix $\boldsymbol{Q}$ with columns $(\boldsymbol{\alpha}(i), 1)$

2. **Rounding step** $\hat{\boldsymbol{\nu}} \leftarrow \texttt{round}\left(\tilde{\boldsymbol{\nu}}, \hat{\delta}\right)$, $\hat{\boldsymbol{c}} \leftarrow \texttt{round}\left(\tilde{\boldsymbol{c}}, \hat{\delta}\right)$
3. **Projection step**
   $\boldsymbol{\nu}^{(j)} \leftarrow (\boldsymbol{I} - \boldsymbol{Q}^{+}\boldsymbol{Q})\, \hat{\boldsymbol{\nu}}^{(j)}\,, \quad \boldsymbol{c}^{(j)}_{\setminus j} \leftarrow \hat{\boldsymbol{c}}^{(j)}_{\setminus j}\,, \quad c^{(j)}_j \leftarrow b_j - \mathbf{1} \cdot \boldsymbol{c}^{(j)}_{\setminus j}$

Algorithm optsage: symbolic steps

- **Input** $f = \sum_i b_i \exp[\mathbf{x} \cdot \boldsymbol{\alpha}(i)]$, $\hat{\delta}$, $\tilde{\delta}$

Build the matrix $\boldsymbol{Q}$ with columns $(\boldsymbol{\alpha}(i), 1)$

2. **Rounding step** $\hat{\boldsymbol{\nu}} \leftarrow \texttt{round}\left(\tilde{\boldsymbol{\nu}}, \hat{\delta}\right)$, $\hat{\boldsymbol{c}} \leftarrow \texttt{round}\left(\tilde{\boldsymbol{c}}, \hat{\delta}\right)$
3. **Projection step**

$$\boldsymbol{\nu}^{(j)} \leftarrow (\boldsymbol{I} - \boldsymbol{Q}^{+}\boldsymbol{Q})\, \hat{\boldsymbol{\nu}}^{(j)}, \quad \boldsymbol{c}_{\setminus j}^{(j)} \leftarrow \hat{\boldsymbol{c}}_{\setminus j}^{(j)}, \quad c_j^{(j)} \leftarrow b_j - \mathbf{1} \cdot \boldsymbol{c}_{\setminus j}^{(j)}$$

Compute $c_1^{(j)}$ such that

$$c_j^{(j)} \geqslant D\left(\boldsymbol{\nu}_{\setminus j}^{(j)}, e\boldsymbol{c}_{\setminus j}^{(j)}\right) = \sum_{j \neq i > 1} \nu_i^{(j)} \log \frac{\nu_i^{(j)}}{ec_i^{(j)}} + \nu_1^{(j)} \log \frac{\nu_1^{(j)}}{ec_1^{(j)}}$$

Algorithm optsage: symbolic steps

- **Input** $f = \sum_i b_i \exp[\mathbf{x} \cdot \boldsymbol{\alpha}(i)]$, $\hat{\delta}$, $\tilde{\delta}$

Build the matrix $\boldsymbol{Q}$ with columns $(\boldsymbol{\alpha}(i), 1)$

2. **Rounding step** $\hat{\boldsymbol{\nu}} \leftarrow \texttt{round}\left(\tilde{\boldsymbol{\nu}}, \hat{\delta}\right)$, $\hat{\boldsymbol{c}} \leftarrow \texttt{round}\left(\tilde{\boldsymbol{c}}, \hat{\delta}\right)$
3. **Projection step**

$$\boldsymbol{\nu}^{(j)} \leftarrow (\boldsymbol{I} - \boldsymbol{Q}^{+}\boldsymbol{Q})\,\hat{\boldsymbol{\nu}}^{(j)}, \quad \boldsymbol{c}_{\setminus j}^{(j)} \leftarrow \hat{\boldsymbol{c}}_{\setminus j}^{(j)}, \quad c_j^{(j)} \leftarrow b_j - \mathbf{1} \cdot \boldsymbol{c}_{\setminus j}^{(j)}$$

Compute $c_1^{(j)}$ such that

$$c_j^{(j)} \geqslant D\left(\boldsymbol{\nu}_{\setminus j}^{(j)}, e\boldsymbol{c}_{\setminus j}^{(j)}\right) = \sum_{j \neq i > 1} \nu_i^{(j)} \log \frac{\nu_i^{(j)}}{ec_i^{(j)}} + \nu_1^{(j)} \log \frac{\nu_1^{(j)}}{ec_1^{(j)}}$$

$$c_1^{(j)} \leftarrow \texttt{round}\uparrow\left(\exp(\ldots), \hat{\delta}\right)$$

Algorithm `optsage`: symbolic steps

- **Input** $f = \sum_i b_i \exp[\mathbf{x} \cdot \boldsymbol{\alpha}(i)]$, $\hat{\delta}$, $\tilde{\delta}$

Build the matrix $\boldsymbol{Q}$ with columns $(\boldsymbol{\alpha}(i), 1)$

2. **Rounding step** $\hat{\boldsymbol{\nu}} \leftarrow \mathtt{round}\left(\tilde{\boldsymbol{\nu}}, \hat{\delta}\right)$, $\hat{\boldsymbol{c}} \leftarrow \mathtt{round}\left(\tilde{\boldsymbol{c}}, \hat{\delta}\right)$
3. **Projection step**

$$\boldsymbol{\nu}^{(j)} \leftarrow (\boldsymbol{I} - \boldsymbol{Q}^{+}\boldsymbol{Q})\, \hat{\boldsymbol{\nu}}^{(j)}, \quad \boldsymbol{c}^{(j)}_{\backslash j} \leftarrow \hat{\boldsymbol{c}}^{(j)}_{\backslash j}, \quad c^{(j)}_{j} \leftarrow b_j - \mathbf{1} \cdot \boldsymbol{c}^{(j)}_{\backslash j}$$

Compute $c_1^{(j)}$ such that

$$c_j^{(j)} \geqslant D\left(\boldsymbol{\nu}^{(j)}_{\backslash j}, e\boldsymbol{c}^{(j)}_{\backslash j}\right) = \sum_{j \neq i > 1} \nu_i^{(j)} \log \frac{\nu_i^{(j)}}{e c_i^{(j)}} + \nu_1^{(j)} \log \frac{\nu_1^{(j)}}{e c_1^{(j)}}$$

$$c_1^{(j)} \leftarrow \mathtt{round} \uparrow \left(\exp(\ldots), \hat{\delta}\right)$$

$$\boxed{f \geqslant b_1 - \sum_j c_1^{(j)}}$$

Univariate SOS

Multivariate SOS

SONC/SAGE

Benchmarks

SOS vs CAD & critical points

- rounding-projection (SOS) [Peyrl-Parrilo]
- `RAGLib` (critical points) [Safey El Din]
- `SamplePoints` (CAD) [Moreno Maza-Alvandi et al.]

Id	n	d	`RealCertify` τ_1 (bits)	`RealCertify` t_1 (s)	`RoundProject` τ_2 (bits)	`RoundProject` t_2 (s)	`RAGLib` t_3 (s)	`CAD` t_4 (s)
f_{20}	2	20	745 419	110.	78 949 497	141.	0.16	0.03
M	3	8	17 232	0.35	18 831	0.29	0.15	0.03
f_2	2	4	1 866	0.03	1 031	0.04	0.09	0.01
f_6	6	4	56 890	0.34	475 359	0.54	598.	–
f_{10}	10	4	344 347	2.45	8 374 082	4.59	–	–

SONC vs SAGE

terms	bit size		time	
	`optsonc`	`optsage`	`optsonc`	`optsage`
6	432	1005	0.06	0.26
9	806	2696	0.19	0.66
12	1261	5568	0.37	1.29
20	2592	19203	0.64	4.00
24	3826	32543	0.97	6.66
30	5029	53160	1.34	10.58
50	10622	167971	3.95	32.78

SONC vs SAGE

terms	bit size		time	
	`optsonc`	`optsage`	`optsonc`	`optsage`
6	432	1005	0.06	0.26
9	806	2696	0.19	0.66
12	1261	5568	0.37	1.29
20	2592	19203	0.64	4.00
24	3826	32543	0.97	6.66
30	5029	53160	1.34	10.58
50	10622	167971	3.95	32.78

"IN PRACTICE" `optsonc` faster and more concise than `optsage`

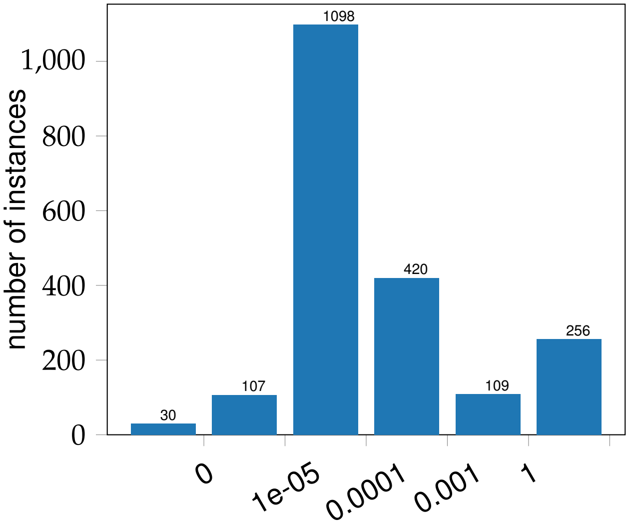
SONC vs SAGE

terms	bit size		time	
	`optsonc`	`optsage`	`optsonc`	`optsage`
6	432	1005	0.06	0.26
9	806	2696	0.19	0.66
12	1261	5568	0.37	1.29
20	2592	19203	0.64	4.00
24	3826	32543	0.97	6.66
30	5029	53160	1.34	10.58
50	10622	167971	3.95	32.78

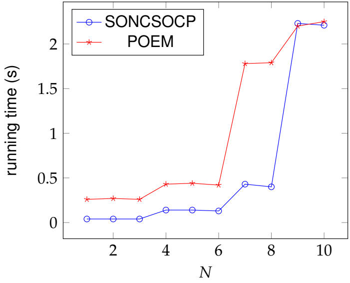
“IN PRACTICE” `optsonc` faster and more concise than `optsage`

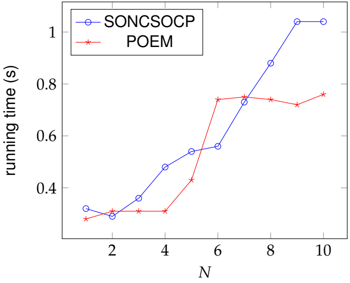
“IN THEORY” `optsonc` less accurate than `optsage`

SONC: gap between numeric & symbolic



SONC: SOCP vs GP

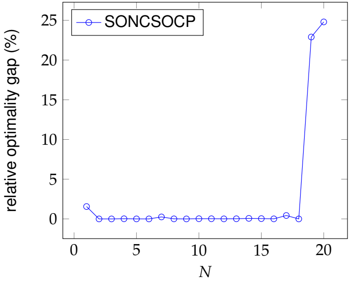




Arbitrary support

$n \sim 40$

$d \sim 60$



Conclusion and Perspectives

Input f on $\mathbf{K}$ with $\deg f = d$, t terms and bit size τ

Algo	Input	$\mathbf{K}$	COMPLEXITY

Conclusion and Perspectives

Input f on $\mathbf{K}$ with $\deg f = d$, t terms and bit size τ

Algo	Input	$\mathbf{K}$	COMPLEXITY
`univsos`	Σ	$\mathbb{R}$	$\widetilde{\mathcal{O}}\left(d^4 + d^3\tau\right)$

Conclusion and Perspectives

Input f on $\mathbf{K}$ with $\deg f = d$, t terms and bit size τ

Algo	Input	$\mathbf{K}$	Complexity
`univsos`	Σ	$\mathbb{R}$	$\widetilde{\mathcal{O}}\left(d^4 + d^3\tau\right)$
`csos`	$\mathring{\Sigma}$	$\mathbb{S}^1$	$\widetilde{\mathcal{O}}\left(d^6(d + \tau)\right)$

Conclusion and Perspectives

Input f on $\mathbf{K}$ with $\deg f = d$, t terms and bit size τ

Algo	Input	$\mathbf{K}$	Complexity
`univsos`	Σ	$\mathbb{R}$	$\widetilde{\mathcal{O}}\left(d^4 + d^3\tau\right)$
`csos`	$\mathring{\Sigma}$	$\mathbb{S}^1$	$\widetilde{\mathcal{O}}\left(d^6(d+\tau)\right)$
`multivsos`	$\mathring{\Sigma}$	$\mathbb{R}^n$	$\tau^2\, d^{d^{\mathcal{O}(n)}}$

Conclusion and Perspectives

Input f on $\mathbf{K}$ with $\deg f = d$, t terms and bit size τ

Algo	Input	$\mathbf{K}$	COMPLEXITY
`univsos`	Σ	$\mathbb{R}$	$\widetilde{\mathcal{O}}\left(d^4 + d^3\tau\right)$
`csos`	$\mathring{\Sigma}$	$\mathbb{S}^1$	$\widetilde{\mathcal{O}}\left(d^6(d+\tau)\right)$
`multivsos`	$\mathring{\Sigma}$	$\mathbb{R}^n$	$\tau^2\, d^{d^{\mathcal{O}(n)}}$
`gradsos`	$\geqslant 0$	$V_{\text{grad}}(f)$	$\widetilde{\mathcal{O}}\left((\tau + n + d)d^{4n+4}\right)$

Conclusion and Perspectives

Input f on $\mathbf{K}$ with $\deg f = d$, t terms and bit size τ

Algo	Input	$\mathbf{K}$	COMPLEXITY
`univsos`	Σ	$\mathbb{R}$	$\widetilde{\mathcal{O}}\left(d^4 + d^3\tau\right)$
`csos`	$\mathring{\Sigma}$	$\mathbb{S}^1$	$\widetilde{\mathcal{O}}\left(d^6(d+\tau)\right)$
`multivsos`	$\mathring{\Sigma}$	$\mathbb{R}^n$	$\tau^2\, d^{d^{\mathcal{O}(n)}}$
`gradsos`	$\geqslant 0$	$V_{\text{grad}}(f)$	$\widetilde{\mathcal{O}}\left((\tau + n + d)d^{4n+4}\right)$
`intsage`	$\mathring{\mathcal{C}}_{\mathsf{SAGE}}$	$\mathbb{R}^n$	$\widetilde{\mathcal{O}}\left(\tau_{\mathsf{SAGE}}(f) + \tau + t)t^7\right)$

Conclusion and Perspectives

Input f on $\mathbf{K}$ with $\deg f = d$, t terms and bit size τ

Algo	Input	$\mathbf{K}$	Complexity
`univsos`	Σ	$\mathbb{R}$	$\widetilde{\mathcal{O}}\left(d^4 + d^3\tau\right)$
`csos`	$\mathring{\Sigma}$	$\mathbb{S}^1$	$\widetilde{\mathcal{O}}\left(d^6(d+\tau)\right)$
`multivsos`	$\mathring{\Sigma}$	$\mathbb{R}^n$	$\tau^2\, d^{d^{\mathcal{O}(n)}}$
`gradsos`	$\geqslant 0$	$V_{\text{grad}}(f)$	$\widetilde{\mathcal{O}}\left((\tau + n + d)d^{4n+4}\right)$
`intsage`	$\mathring{\mathcal{C}}_{\text{SAGE}}$	$\mathbb{R}^n$	$\widetilde{\mathcal{O}}\left(\tau_{\text{SAGE}}(f) + \tau + t)t^7\right)$
`intsonc`	$\mathring{\mathcal{C}}_{\text{SONC}}$	$\mathbb{R}^n$	$\widetilde{\mathcal{O}}\left(\tau_{\text{SONC}}(g)d^{3.5n}n^{10.5})\right)$

Conclusion and Perspectives

Input f on $\mathbf{K}$ with $\deg f = d$, t terms and bit size τ

Algo	Input	$\mathbf{K}$	Complexity
univsos	Σ	$\mathbb{R}$	$\widetilde{\mathcal{O}}\left(d^4 + d^3\tau\right)$
csos	$\mathring{\Sigma}$	$\mathbb{S}^1$	$\widetilde{\mathcal{O}}\left(d^6(d+\tau)\right)$
multivsos	$\mathring{\Sigma}$	$\mathbb{R}^n$	$\tau^2\, d^{d^{\mathcal{O}(n)}}$
gradsos	$\geqslant 0$	$V_{\text{grad}}(f)$	$\widetilde{\mathcal{O}}\left((\tau + n + d)d^{4n+4}\right)$
intsage	$\mathring{\mathcal{C}}_{\text{SAGE}}$	$\mathbb{R}^n$	$\widetilde{\mathcal{O}}\left(\tau_{\text{SAGE}}(f) + \tau + t)t^7\right)$
intsonc	$\mathring{\mathcal{C}}_{\text{SONC}}$	$\mathbb{R}^n$	$\widetilde{\mathcal{O}}\left(\tau_{\text{SONC}}(g)d^{3.5n}n^{10.5})\right)$

- How to bound the distance to the SONC/SAGE cone?
- Arbitrary precision SDP/GP/REP/SOCP solvers
- How to handle degenerate situations?

Thank you for your (certified) attention!

RealCertify POEM SONCSOCP

https://homepages.laas.fr/vmagron

- M., Allamigeon, Gaubert & Werner. Formal proofs for Nonlinear Optimization, *Journal of Formalized Reasoning*. arXiv:1404.7282
- M. & Safey El Din. On Exact Polya and Putinar's Representations, *ISSAC'18*. arXiv:1802.10339 RealCertify
- M. & Safey El Din. RealCertify: a Maple package for certifying non-negativity, *ISSAC'18*. arXiv:1805.02201 RealCertify
- M., Safey El Din & Schweighofer. Algorithms for weighted sum of squares decomposition of non-negative univariate polynomials, *Journal of Symbolic Computation*. arXiv:1706.03941
- M., Seidler & de Wolff. Exact optimization via sums of nonnegative circuits and AM/GM exponentials, *ISSAC'19*. arXiv:1902.02123 POEM
- M. & Wang. SONC Optimization & Exact Nonnegativity Certificates via Second-order Conic Programming, *ISSAC'20*. arXiv:2012.07903 SONCSOCP
- M., Safey El Din & Vu. Sum of Squares Decompositions of Polynomials over their Gradient Ideals with Rational Coefficients. arXiv:2107.11825
- M., Safey El Din, Schweighofer, & Vu. Exact SOHS decompositions of trigonometric univariate polynomials with Gaussian coefficients. arXiv:2202.06544