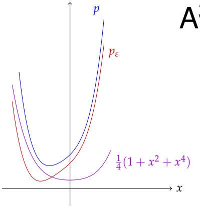
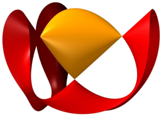
Exact Polynomial Optimization via SOS, SONC and SAGE Decompositions

Victor Magron, LAAS CNRS

Joint work with
Mohab Safey El Din (Sorbonne Université)
Markus Schweighofer (Konstanz University)
Timo de Wolff (TU Braunschweig)
Henning Seidler (TU Berlin)

A^3 - Arctic Applied Algebra, Tromsø
1st April 2019





Deciding Nonnegativity & Exact Optimization

$X = (X_1, \dots, X_n)$ **co-NP hard problem: check** $f \geqslant 0$ **on** $\mathbf{K}$

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

Deciding Nonnegativity & Exact Optimization

$X = (X_1, \ldots, X_n)$ **co-NP hard problem: check $f \geqslant 0$ on $\mathbf{K}$**

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

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

2. Constrained
   $\rightsquigarrow \mathbf{K} = \{\mathbf{x} \in \mathbb{R}^n : g_1(\mathbf{x}) \geqslant 0, \ldots, g_m(\mathbf{x}) \geqslant 0\} \quad g_j \in \mathbb{Q}[X]$

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

Deciding Nonnegativity & Exact Optimization

$X = (X_1, \dots, X_n)$ **co-NP hard problem: check** $f \geqslant 0$ **on** $\mathbf{K}$

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

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

2. Constrained
   $\rightsquigarrow \mathbf{K} = \{\mathbf{x} \in \mathbb{R}^n : g_1(\mathbf{x}) \geqslant 0, \dots, g_m(\mathbf{x}) \geqslant 0\}$ $\quad g_j \in \mathbb{Q}[X]$

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

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

[Grigoriev-Vorobjov 88, Basu-Pollack-Roy 98]
Critical points **singly exponential time** $(m+1)\,\tau\, d^{\mathcal{O}(n)}$

Deciding Nonnegativity & Exact Optimization

Sums of squares (SOS)

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

Deciding Nonnegativity & Exact Optimization

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 & Exact Optimization

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 & Exact Optimization

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)

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

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

Decomposing Nonnegative Polynomials

4. **Circuit** polynomial

$$f = b_{\boldsymbol{\alpha}(1)} X^{\boldsymbol{\alpha}(1)} + \cdots + b_{\boldsymbol{\alpha}(r)} X^{\boldsymbol{\alpha}(r)} + b_{\boldsymbol{\beta}} X^{\boldsymbol{\beta}}$$

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

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

Decomposing Nonnegative Polynomials

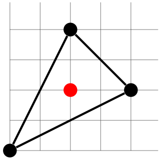
4. **Circuit** polynomial

$$f = b_{\alpha(1)} X^{\alpha(1)} + \cdots + b_{\alpha(r)} X^{\alpha(r)} + b_\beta X^\beta$$

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

$\beta = \lambda_1 \alpha(1) + \cdots + \lambda_r \alpha(r) \quad \lambda_j > 0$ and $\lambda_1 + \cdots + \lambda_r = 1$

$$f = 1 + X_1^2 X_2^4 + X_1^4 X_2^2 - 3 X_1^2 X_2^2$$



Decomposing Nonnegative Polynomials

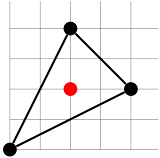
4 **Circuit** polynomial

$f = b_{\alpha(1)} X^{\alpha(1)} + \cdots + b_{\alpha(r)} X^{\alpha(r)} + b_{\beta} X^{\beta}$

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

$\beta = \lambda_1 \alpha(1) + \cdots + \lambda_r \alpha(r) \quad \lambda_j > 0$ and $\lambda_1 + \cdots + \lambda_r = 1$

$$f = 1 + X_1^2 X_2^4 + X_1^4 X_2^2 - 3 X_1^2 X_2^2$$



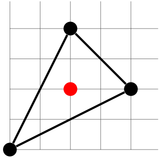
Circuit number $\Theta_f = \prod_{j=1}^{r} \left(\frac{b_{\alpha(j)}}{\lambda_j} \right)^{\lambda_j}$

Decomposing Nonnegative Polynomials

4. **Circuit** polynomial
$f = b_{\alpha(1)} X^{\alpha(1)} + \cdots + b_{\alpha(r)} X^{\alpha(r)} + b_{\beta} X^{\beta}$

$b_{\alpha(j)} > 0 \quad \alpha(j) \in (2\mathbb{N})^n$
$\beta = \lambda_1 \alpha(1) + \cdots + \lambda_r \alpha(r) \quad \lambda_j > 0$ and $\lambda_1 + \cdots + \lambda_r = 1$

$$f = 1 + X_1^2 X_2^4 + X_1^4 X_2^2 - 3 X_1^2 X_2^2$$



Circuit number $\Theta_f = \prod_{j=1}^{r} \left(\frac{b_{\alpha(j)}}{\lambda_j} \right)^{\lambda_j}$

Theorem (Illiman-de Wolff 16)

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

SONC (SUMS OF NONNEGATIVE CIRCUITS)

Decomposing Nonnegative Polynomials

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

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

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

Decomposing Nonnegative Polynomials

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

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

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

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

Decomposing Nonnegative Polynomials

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

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

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

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

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)

From Approximate to Exact Solutions

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

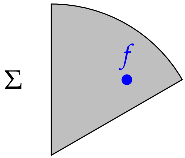
Does there exist $f_i \in \mathbb{Q}[X], c_i \in \mathbb{Q}^{>0}$ s.t. $f = \sum_i c_i f_i{}^2$?

Examples

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

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

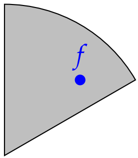
Round & Project Algorithm [Peyrl-Parrilo 08]



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

Round & Project Algorithm [Peyrl-Parrilo 08]

Σ



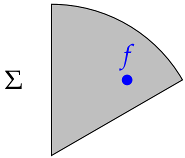
$f \in \mathring{\Sigma}[X]$ with $\deg f = 2D$

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

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

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

Round & Project Algorithm [Peyrl-Parrilo 08]



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

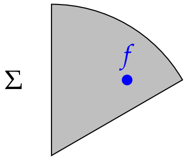
Find $\tilde{\mathbf{G}}$ with SDP at tolerance $\tilde{\delta}$ satisfying

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

$\mathbf{v}_D(X)$: 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]



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

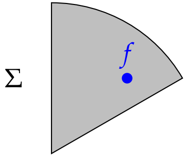
Find $\tilde{\mathbf{G}}$ with SDP at tolerance $\tilde{\delta}$ satisfying

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

$\mathbf{v}_D(X)$: 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]



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

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

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

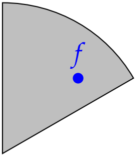
$\mathbf{v}_D(X)$: vector of monomials of $\deg \leqslant D$

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



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

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

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

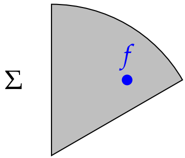
$\mathbf{v}_D(X)$: vector of monomials of $\deg \leqslant D$

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]



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

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

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

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

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

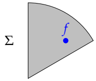
1. **Rounding step** $\hat{\boldsymbol{G}} \leftarrow \texttt{round}\left(\tilde{\boldsymbol{G}}, \hat{\delta}\right)$
2. **Projection step**

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

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

From Approximate to Exact Solutions

Win TWO-PLAYER GAME

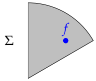


sum of squares of f?

≃ Output!

From Approximate to Exact Solutions

Win TWO-PLAYER GAME



Hybrid Symbolic/Numeric Algorithms

sum of squares of $f + \varepsilon$?

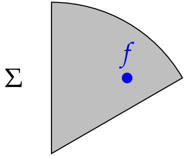
$\simeq$ Output!

Error Compensation

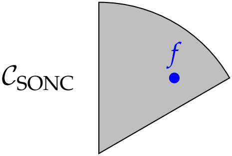
$\simeq \quad \to \quad =$

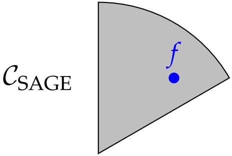
From Approximate to Exact Solutions

Exact SOS



Exact SONC/SAGE





Software: RealCertify and POEM

Exact optimization via SOS: RealCertify

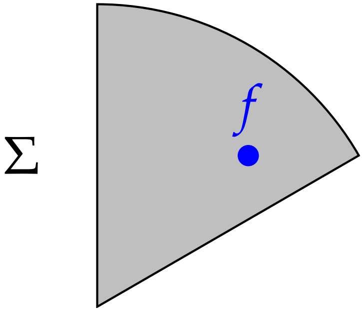
Maple & arbitrary precision SDP solver SDPA-GMP [Nakata 10]

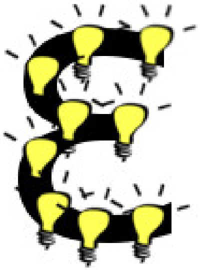
univsos	$n = 1$
multivsos	$n > 1$

Exact optimization via SONC/SAGE: POEM

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

intsos with $n \geqslant 1$: Perturbation





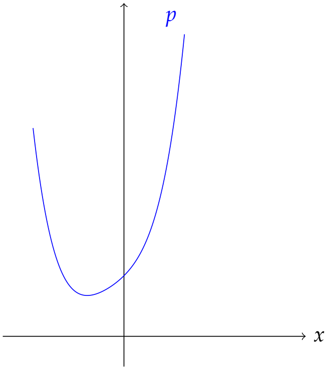
Perturbation idea

Approximate SOS Decomposition

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

intsos with $n = 1$ [Chevillard et. al 11]

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



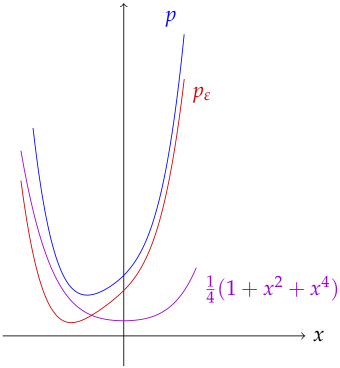
$p = 1 + X + X^2 + X^3 + X^4$

intsos with $n = 1$ [Chevillard et. al 11]

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

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

$$p_\varepsilon := p - \varepsilon \sum_{i=0}^{k} X^{2i} > 0$$



$$p = 1 + X + X^2 + X^3 + X^4$$

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

$$p > \frac{1}{4}(1 + X^2 + X^4)$$

intsos with $n = 1$ [Chevillard et. al 11]

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

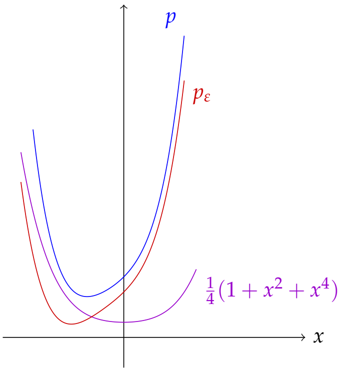
PERTURB: find $\varepsilon \in \mathbb{Q}$ s.t.

$$p_\varepsilon := p - \varepsilon \sum_{i=0}^{k} X^{2i} > 0$$

SDP Approximation:

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

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



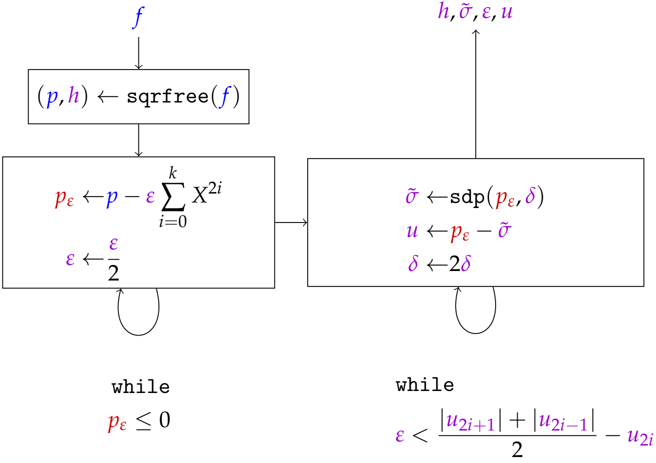
$$p = 1 + X + X^2 + X^3 + X^4$$

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

$$p > \frac{1}{4}(1 + X^2 + X^4)$$

intsos with $n = 1$ and SDP Approximation

- **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 $\mathbb{Q}$



`intsos` with $n = 1$: Absorbtion

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

intsos with $n = 1$: Absorbtion

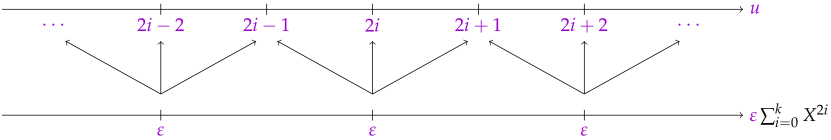
- $X = \frac{1}{2}\left[(X+1)^2 - 1 - X^2\right]$
- $-X = \frac{1}{2}\left[(X-1)^2 - 1 - X^2\right]$

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

intsos with $n = 1$: Absorbtion

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

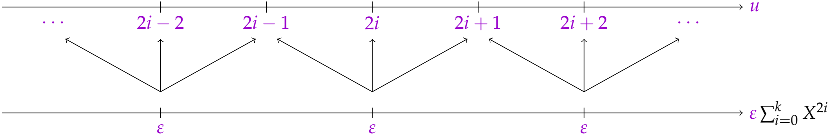
$$u_{2i+1} X^{2i+1} = \frac{|u_{2i+1}|}{2}\left[(X^{i+1} + \operatorname{sgn}(u_{2i+1}) X^i)^2 - X^{2i} - X^{2i+2}\right]$$



intsos with $n = 1$: Absorbtion

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

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

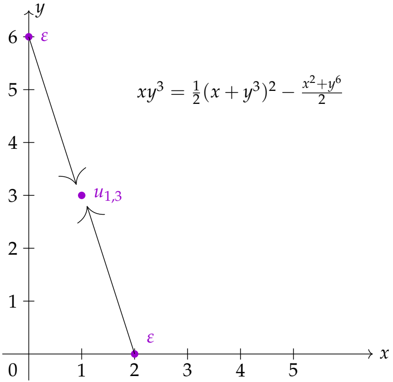


$$\varepsilon \geqslant \frac{|u_{2i+1}| + |u_{2i-1}|}{2} - u_{2i} \implies \varepsilon \sum_{i=0}^{k} X^{2i} + u \quad \text{SOS}$$

$\texttt{intsos}$ with $n \geqslant 1$: Absorbtion

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

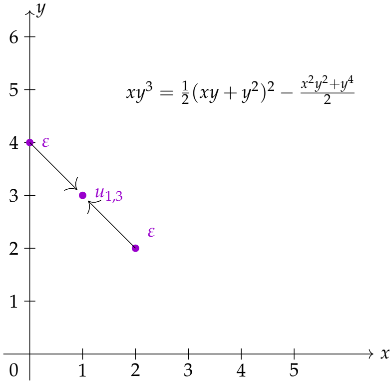
Choice of $\mathcal{P}$?



intsos with $n \geqslant 1$: Absorbtion

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

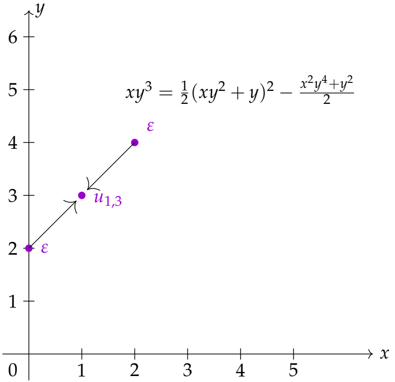
Choice of $\mathcal{P}$?



intsos with $n \geqslant 1$: Absorbtion

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

Choice of $\mathcal{P}$?

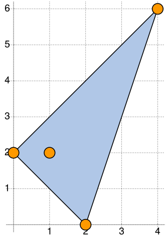


intsos with $n \geqslant 1$: Absorbtion

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

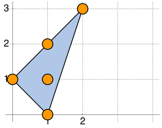
Choice of $\mathcal{P}$?

$f = 4x^4y^6 + x^2 - xy^2 + y^2$
spt(f) = $\{(4,6),(2,0),(1,2),(0,2)\}$



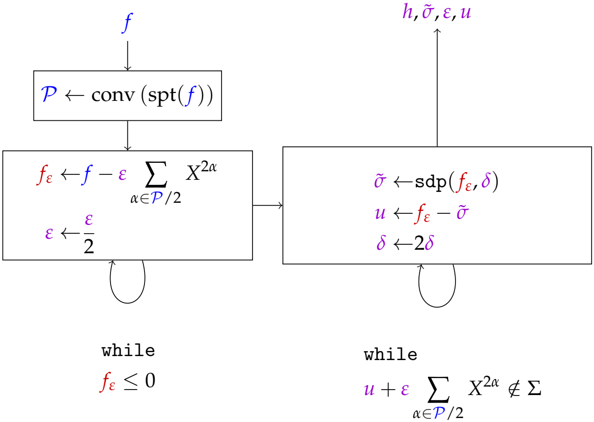
Newton Polytope $\mathcal{P} = \mathrm{conv}\,(\text{spt}(f))$

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



Algorithm `intsos`

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



Algorithm `optsonc`: numerical steps

SONC (SUMS OF NONNEGATIVE CIRCUITS)

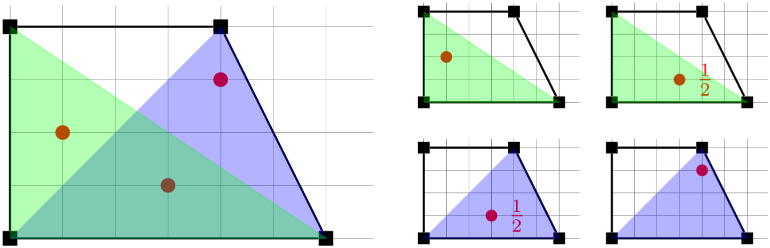
- **Input** $f = \sum_{\boldsymbol{\alpha}} b_{\boldsymbol{\alpha}} X^{\boldsymbol{\alpha}}$ of degree d, $\hat{\delta} \in \mathbb{Q}^{>0}$, $\tilde{\delta} \in \mathbb{Q}^{>0}$
  Monomial squares = $\mathrm{MoSq}\,(f)$ Complement = $\mathrm{NoSq}\,(f)$

Algorithm optsonc: numerical steps

SONC (SUMS OF NONNEGATIVE CIRCUITS)

- **Input** $f = \sum_{\alpha} b_{\alpha} X^{\alpha}$ of degree d, $\hat{\delta} \in \mathbb{Q}^{>0}$, $\tilde{\delta} \in \mathbb{Q}^{>0}$
  Monomial squares = $\mathrm{MoSq}(f)$ Complement = $\mathrm{NoSq}(f)$

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



Algorithm `optsonc`: numerical steps

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

2. **Numerical** resolution of GEOMETRIC PROGRAM

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

Algorithm `optsonc`: symbolic steps

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

GEOMETRIC PROGRAM provides "IN THEORY"

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

Algorithm `optsonc`: symbolic steps

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

GEOMETRIC PROGRAM provides "IN THEORY"

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

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

Algorithm `optsonc`: symbolic steps

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

GEOMETRIC PROGRAM provides "IN THEORY"

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

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** $f = \sum_{\alpha} b_{\alpha} X^{\alpha}$, $\hat{\delta}$, $\tilde{\delta}$

GEOMETRIC PROGRAM provides "IN THEORY"

$$f_{\beta} = \sum_{\alpha \in \mathrm{Cov}^{\beta}} G_{\beta,\alpha} \cdot X^{\alpha} + b_{\beta} X^{\beta}, \; f + \sum_{\beta} G_{\beta,0} - b_0 = \sum_{\beta} f_{\beta} \geqslant 0$$

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**

$$G_{\beta,\alpha} \leftarrow b_{\alpha} \cdot \hat{G}_{\beta,\alpha} / \sum_{\beta' \in \mathrm{NoSq}(f)} \hat{G}_{\beta',\alpha}$$

$$\tilde{G}_{\beta,0} \leftarrow \lambda_0^{\beta} \left(-b_{\beta} \cdot \prod_{\alpha \in \mathrm{Cov}^{\beta}} \left(\frac{\lambda_{\alpha}^{\beta}}{G_{\beta,\alpha}} \right)^{\lambda_{\alpha}^{\beta}} \right)^{\frac{1}{\lambda_0^{\beta}}}$$

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

Algorithm optsonc: symbolic steps

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

GEOMETRIC PROGRAM provides "IN THEORY"

$$f_{\beta} = \sum_{\alpha \in \mathrm{Cov}^{\beta}} G_{\beta,\alpha} \cdot X^{\alpha} + b_{\beta} X^{\beta}, \; f + \sum_{\beta} G_{\beta,0} - b_0 = \sum_{\beta} f_{\beta} \geqslant 0$$

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

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

$$G_{\beta,\alpha} \leftarrow b_{\alpha} \cdot \hat{G}_{\beta,\alpha} / \sum_{\beta' \in \mathrm{NoSq}(f)} \hat{G}_{\beta',\alpha}$$

$$\tilde{G}_{\beta,0} \leftarrow \lambda_0^{\beta} \left(-b_{\beta} \cdot \prod_{\alpha \in \mathrm{Cov}^{\beta}} \left(\frac{\lambda_{\alpha}^{\beta}}{G_{\beta,\alpha}} \right)^{\lambda_{\alpha}^{\beta}} \right)^{\frac{1}{\lambda_0^{\beta}}}$$

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

$$\boxed{f \geqslant b_0 - \sum_{\beta} \hat{G}_{\beta,0}}$$

Algorithm `optsage`: numerical steps

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

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

Algorithm `optsage`: numerical steps

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

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

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

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

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

Algorithm optsage: numerical steps

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

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

$f \in \mathcal{C}_{\text{SAGE}} \Leftrightarrow \exists \boldsymbol{\nu}^{(j)}, \boldsymbol{c}^{(j)}$ 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** $f = \sum_i b_i \exp[X \cdot \boldsymbol{\alpha}(i)]$, $\hat{\delta}$, $\tilde{\delta}$

Algorithm `optsage`: symbolic steps

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

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

Algorithm optsage: symbolic steps

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

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

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

Algorithm `optsage`: symbolic steps

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

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

Algorithm optsage: symbolic steps

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

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**

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

Compute $c_j^{(1)}$ 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** $f = \sum_i b_i \exp[X \cdot \boldsymbol{\alpha}(i)]$, $\hat{\delta}$, $\tilde{\delta}$

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

   Compute $c^{(1)}_j$ such that

   $c^{(j)}_j \geqslant D\left(\boldsymbol{\nu}^{(j)}_{\backslash j}, e\boldsymbol{c}^{(j)}_{\backslash j}\right) = \sum_{j \neq i > 1} \nu^{(j)}_i \log \frac{\nu^{(j)}_i}{e c^{(j)}_i} + \nu^{(j)}_1 \log \frac{\nu^{(j)}_1}{e c^{(j)}_1}$

   $c^{(j)}_1 \leftarrow \texttt{round} \uparrow \left(\exp(\ldots), \hat{\delta}\right)$

Algorithm `optsage`: symbolic steps

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

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**

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

Compute $c^{(1)}_j$ such that

$$c^{(j)}_j \geqslant D\left(\boldsymbol{\nu}^{(j)}_{\backslash j}, e\boldsymbol{c}^{(j)}_{\backslash j}\right) = \sum_{j \neq i > 1} \nu^{(j)}_i \log \frac{\nu^{(j)}_i}{ec^{(j)}_i} + \nu^{(j)}_1 \log \frac{\nu^{(j)}_1}{ec^{(j)}_1}$$

$$c^{(j)}_1 \leftarrow \texttt{round}\uparrow\left(\exp(\ldots), \hat{\delta}\right)$$

$$\boxed{f \geqslant b_1 - \sum_j c^{(j)}_1}$$

SOS Benchmarks

- 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_1	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`

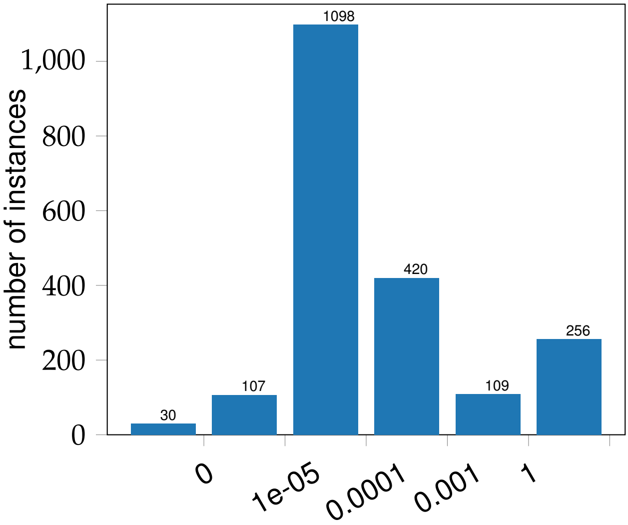
SONC vs SAGE

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



Conclusion and Perspectives

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

Algo	Input	$\mathbf{K}$	Output Bit Size
`intsos`	$\mathring{\Sigma}$	$\mathbb{R}^n$	$\tau\, d^{d^{\mathcal{O}(n)}}$
`intsage`	$\mathring{\mathcal{C}}_{\text{SAGE}}$	$\mathbb{R}^n$	

Conclusion and Perspectives

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

Algo	Input	$\mathbf{K}$	Output Bit Size
intsos	$\mathring{\Sigma}$	$\mathbb{R}^n$	$\tau \, d^{d^{\mathcal{O}(n)}}$
intsage	$\mathring{\mathcal{C}}_{\text{SAGE}}$	$\mathbb{R}^n$	

- How to handle degenerate situations?
- Arbitrary precision SDP/GP/REP solvers
- Extension to other relaxations

Crucial need for polynomial systems certification
Available PhD/Postdoc Positions

Thank you for your attention!

RealCertify   POEM

https://homepages.laas.fr/vmagron

- M., Safey El Din & Schweighofer. Algorithms for Weighted Sums of Squares Decomposition of Non-negative Univariate Polynomials, *JSC*. arxiv:1706.03941

- M. & Safey El Din. On Exact Polya and Putinar's Representations, *ISSAC'18*. arxiv:1802.10339

- M. & Safey El Din. RealCertify: a Maple package for certifying non-negativity, *ISSAC'18*. arxiv:1805.02201

- M. & Seidler & de Wolff. Exact optimization via sums of nonnegative circuits and sums of AM/GM exponentials, 2019. arxiv:1902.02123