

# Exact Polynomial Optimization via SOS, SONC and SAGE Decompositions

**Victor Magron**, LAAS CNRS

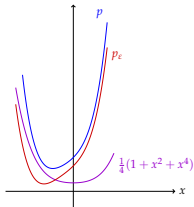
Joint work with

Mohab Safey El Din (Sorbonne Université)

Markus Schweighofer (Konstanz University)

Timo de Wolff (TU Braunschweig)

Henning Seidler (TU Berlin)



EURO 19, Dublin,  
25 June 2019



# Deciding Nonnegativity & Exact Optimization

---

$$X = (X_1, \dots, X_n)$$

**co-NP hard problem: check  $f \geq 0$  on  $\mathbf{K}$**

$$f \in \mathbb{Q}[X]$$

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

# Deciding Nonnegativity & Exact Optimization

---

$$X = (X_1, \dots, X_n)$$

**co-NP hard problem: check  $f \geq 0$  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

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

$$\deg f, \deg g_j \leq d$$

# Deciding Nonnegativity & Exact Optimization

$X = (X_1, \dots, X_n)$  **co-NP hard problem: check  $f \geq 0$  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

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

$\deg f, \deg g_j \leq d$



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



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

# Deciding Nonnegativity & Exact Optimization

---

💡 Sums of squares (SOS)

$$\sigma = h_1^2 + \dots + h_p^2$$

# Deciding Nonnegativity & Exact Optimization

---

💡 Sums of squares (SOS)

$$\sigma = h_1^2 + \dots + h_p^2$$

**HILBERT 17TH PROBLEM:**  $f$  SOS of rational functions?



[Artin 27] **YES!**

# Deciding Nonnegativity & Exact Optimization

---

💡 Sums of squares (SOS)

$$\sigma = h_1^2 + \dots + h_p^2$$

**HILBERT 17TH PROBLEM:**  $f$  SOS of rational functions?



[Artin 27] **YES!**

💡 [Lasserre/Parrilo 01] **Numerical** solvers compute  $\sigma$

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

# Deciding Nonnegativity & Exact Optimization

---

💡 Sums of squares (SOS)

$$\sigma = h_1^2 + \dots + h_p^2$$

**HILBERT 17TH PROBLEM:**  $f$  SOS of rational functions?



[Artin 27] **YES!**

💡 [Lasserre/Parrilo 01] **Numerical** solvers compute  $\sigma$

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

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

## The Question of Exact Certification

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



# Decomposing Nonnegative Polynomials

---

## 1 **Polya's** representation

positive definite form  $f$

[Reznick 95]

$$f = \frac{\sigma}{(X_1^2 + \dots + X_n^2)^D}$$

## 2 **Hilbert-Artin's** representation

$f \geq 0$

[Artin 27]

$$f = \frac{\sigma}{h^2}$$

## 3 **Putinar's** representation

$f = \sigma_0 + \sigma_1 g_1 + \dots + \sigma_m g_m$   $f > 0$  on compact  $K$

$\deg \sigma_i \leq 2D$

[Putinar 93]

# Decomposing Nonnegative Polynomials

---

## ■ Deciding **polynomial nonnegativity**

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

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

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

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

# Decomposing Nonnegative Polynomials

---

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

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

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

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

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

# Decomposing Nonnegative Polynomials

---

## 4 Circuit polynomial

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

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

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

# Decomposing Nonnegative Polynomials

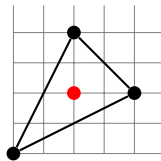
## 4 Circuit polynomial

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

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

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

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



# Decomposing Nonnegative Polynomials

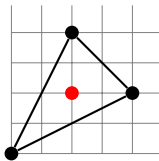
## 4 Circuit polynomial

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

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

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

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



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

# Decomposing Nonnegative Polynomials

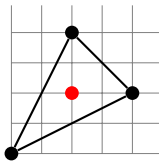
## 4 Circuit polynomial

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

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

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

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



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

Theorem (Illiman-de Wolff 16)

$$f \geq 0 \Leftrightarrow |b_{\beta}| \leq \Theta_f \text{ or } (b_{\beta} \geq -\Theta_f, \beta \text{ even})$$



SONC (SUMS OF NONNEGATIVE CIRCUITS)

# Decomposing Nonnegative Polynomials

---

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

$$f = c_1 \exp[X \cdot \alpha(1)] + \cdots + c_t \exp[X \cdot \alpha(t)] + \beta \exp[X \cdot \alpha(0)]$$
$$c_j \in \mathbb{Q}_{>0} \quad \beta \in \mathbb{Q} \quad \alpha(j) \in \mathbb{N}^n$$



# Decomposing Nonnegative Polynomials

---

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

$$f = c_1 \exp[X \cdot \alpha(1)] + \cdots + c_t \exp[X \cdot \alpha(t)] + \beta \exp[X \cdot \alpha(0)]$$
$$c_j \in \mathbb{Q}_{>0} \quad \beta \in \mathbb{Q} \quad \alpha(j) \in \mathbb{N}^n$$

**relative entropy**       $D(\mathbf{v}, \mathbf{c}) = \sum_j v_j \log \frac{v_j}{c_j}$

# Decomposing Nonnegative Polynomials

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

$$f = c_1 \exp[X \cdot \alpha(1)] + \cdots + c_t \exp[X \cdot \alpha(t)] + \beta \exp[X \cdot \alpha(0)]$$
$$c_j \in \mathbb{Q}_{>0} \quad \beta \in \mathbb{Q} \quad \alpha(j) \in \mathbb{N}^n$$

relative entropy  $D(\mathbf{v}, \mathbf{c}) = \sum_j v_j \log \frac{v_j}{c_j}$

### Theorem (Chandrasekaran-Shah 16)

$$f \geq 0 \Leftrightarrow \exists \mathbf{v} \mid D(\mathbf{v}, \mathbf{c}) \leq \beta \text{ and } \sum_j \alpha(j) v_j = (\mathbf{1} \cdot \mathbf{v}) \alpha(0)$$

💡 SAGE (SUMS OF AGE)

# From Approximate to Exact Solutions

## APPROXIMATE SOLUTIONS

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



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



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

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

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

# Rational SOS Decompositions

---

- Let  $f \in \mathbb{R}[X]$  and  $f \geq 0$  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  $f \geq 0$  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  $f \geq 0$  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

---

- $f \in \mathbb{Q}[X] \cap \overset{\circ}{\Sigma}[X]$  (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

- $f \in \mathbb{Q}[X] \cap \overset{\circ}{\Sigma}[X]$  (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

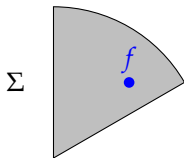
$$1 + X + X^2 = \left(X + \frac{1}{2}\right)^2 + \left(\frac{\sqrt{3}}{2}\right)^2 = 1 \left(X + \frac{1}{2}\right)^2 + \frac{3}{4}(1)^2$$

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



# Round & Project Algorithm [Peyrl-Parrilo 08]

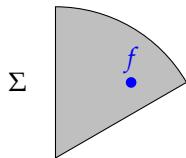
---




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

# Round & Project Algorithm [Peyrl-Parrilo 08]

---



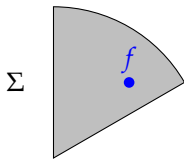
$$f \in \mathring{\Sigma}[X] \text{ with } \deg f = 2D$$

 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  $\deg \leq D$

# Round & Project Algorithm [Peyrl-Parrilo 08]



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

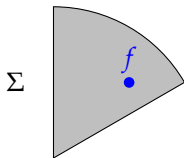
Find  $\tilde{G}$  with SDP at tolerance  $\tilde{\delta}$  satisfying

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

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

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

# Round & Project Algorithm [Peyrl-Parrilo 08]



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

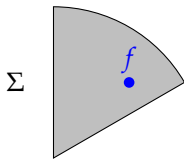
Find  $\tilde{G}$  with SDP at tolerance  $\tilde{\delta}$  satisfying

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

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

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

# Round & Project Algorithm [Peyrl-Parrilo 08]



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

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

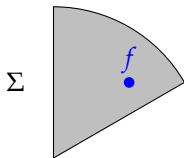
$$f(X) \simeq \mathbf{v}_D^T(X) \tilde{G} \mathbf{v}_D(X) \quad \tilde{G} \succ 0$$

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

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**  $\hat{G} \leftarrow \text{round}(\tilde{G}, \hat{\delta})$

# Round & Project Algorithm [Peyrl-Parrilo 08]



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

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  $\deg \leq D$

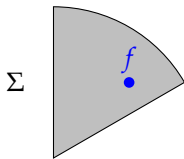
Exact  $G \implies f_\gamma = \sum_{\alpha'+\beta'=\gamma} G_{\alpha',\beta'} \quad f_{\alpha+\beta} = \sum_{\alpha'+\beta'=\alpha+\beta} G_{\alpha',\beta'}$

**1 Rounding step**  $\hat{G} \leftarrow \text{round}(\tilde{G}, \hat{\delta})$

**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]



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

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  $\deg \leq D$

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

**1 Rounding step**  $\hat{\mathbf{G}} \leftarrow \text{round}(\tilde{\mathbf{G}}, \tilde{\delta})$

**2 Projection step**

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

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  $f$ ?



$\approx$  Output!





# From Approximate to Exact Solutions

Win TWO-PLAYER GAME



💡 **Hybrid** Symbolic/Numeric Algorithms

sum of squares of  $f + \varepsilon$ ?

$\approx$  Output!



Error Compensation

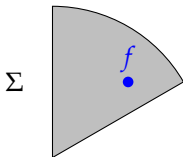
$$\approx \rightarrow =$$



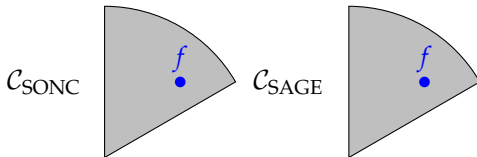
# From Approximate to Exact Solutions

---

Exact SOS



Exact SONC/SAGE



# Software: RealCertify and POEM

---

## Exact optimization via SOS: [RealCertify](#)

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

univsos  $n = 1$

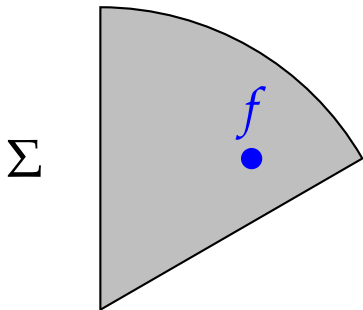
multivsos  $n > 1$

## Exact optimization via SONC/SAGE: [POEM](#)

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

# intsos with $n \geq 1$ : Perturbation

---



## PERTURBATION idea

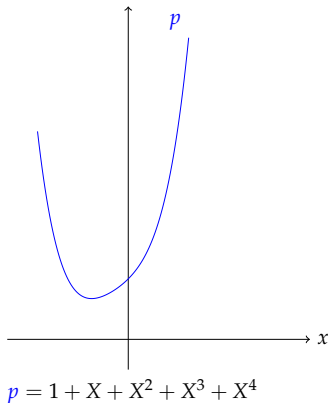
💡 Approximate SOS Decomposition

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

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

---

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

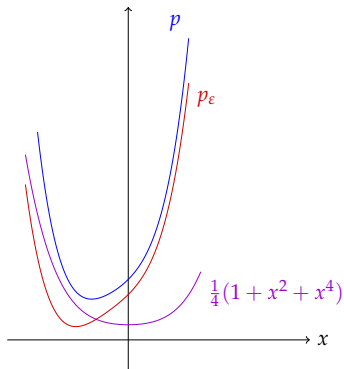


# intsos with $n = 1$ [Chevillard et. al 11]

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

💡 **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.

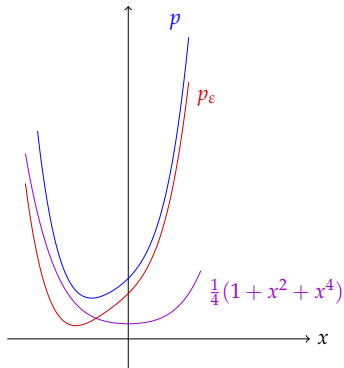
$$p_\varepsilon := p - \varepsilon \sum_{i=0}^k X^{2i} > 0$$

💡 **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



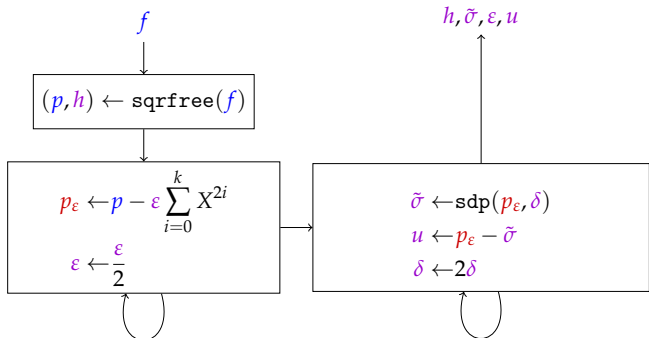
$$p = 1 + X + X^2 + X^3 + X^4$$

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

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

# intsos with $n = 1$ and SDP Approximation

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



while  
 $p_\epsilon \leq 0$

while  
 $\epsilon < \frac{|u_{2i+1}| + |u_{2i-1}|}{2} - u_{2i}$



## intsos with $n = 1$ : Absorbtion

---

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

$$\text{💡 } -X = \frac{1}{2}[(X-1)^2 - 1 - X^2]$$

## intsos with $n = 1$ : Absorbtion

---

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

$$\text{💡 } -X = \frac{1}{2}[(X-1)^2 - 1 - X^2]$$

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

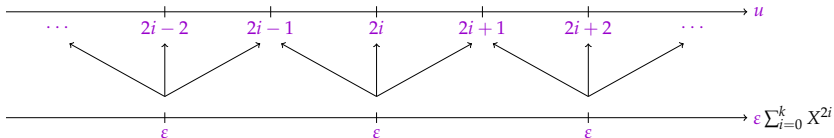
## intsos with $n = 1$ : Absorbion

---

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

$$\text{💡 } -X = \frac{1}{2}[(X-1)^2 - 1 - X^2]$$

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

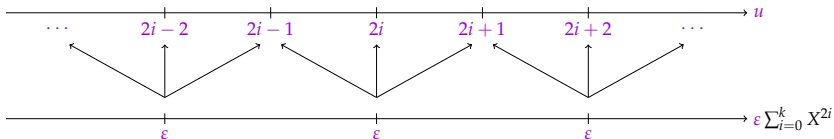


# intsos with $n = 1$ : Absorbion

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

💡  $-X = \frac{1}{2}[(X-1)^2 - 1 - X^2]$

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

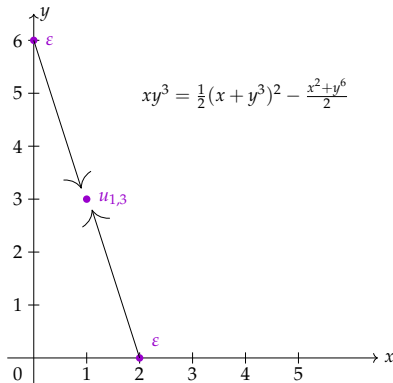


$$\epsilon \geq \frac{|u_{2i+1}| + |u_{2i-1}|}{2} - u_{2i} \implies \epsilon \sum_{i=0}^k X^{2i} + u \quad \text{SOS}$$

# intsos with $n \geq 1$ : Absorbion

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

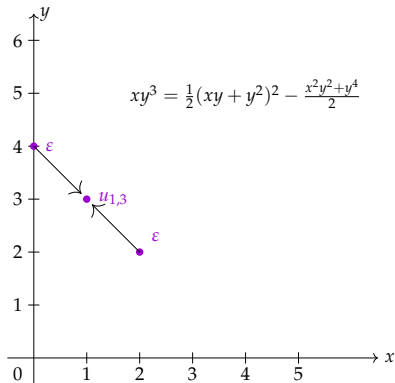
Choice of  $\mathcal{P}$ ?



# intsos with $n \geq 1$ : Absorbion

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

Choice of  $\mathcal{P}$ ?

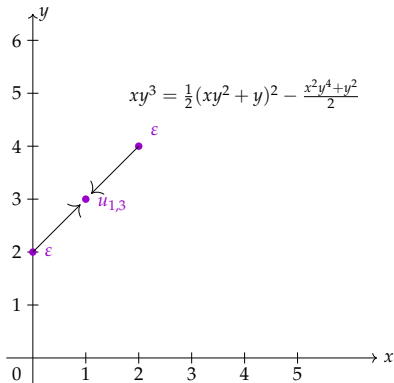


## intsos with $n \geq 1$ : Absorbion

---

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

Choice of  $\mathcal{P}$ ?



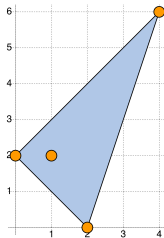
# intsos with $n \geq 1$ : Absorbion

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

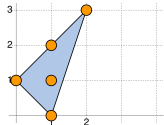
Choice of  $\mathcal{P}$ ?

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

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



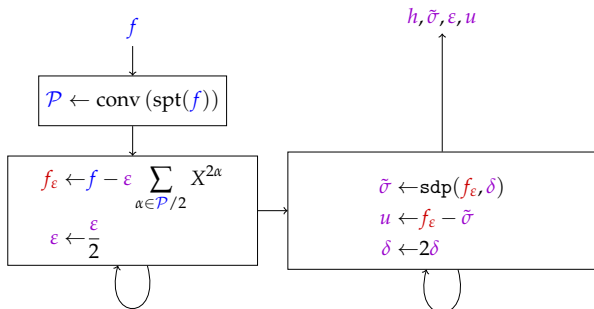
Squares in SOS decomposition  $\subseteq \frac{\mathcal{P}}{2} \cap \mathbb{N}^n$   
[Reznick 78]





# Algorithm intsos

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



while  
 $f_\varepsilon \leq 0$

while  
 $u + \varepsilon \sum_{\alpha \in \mathcal{P}/2} X^{2\alpha} \notin \Sigma$

# Algorithm optsonc: numerical steps

---

## SONC (SUMS OF NONNEGATIVE CIRCUITS)

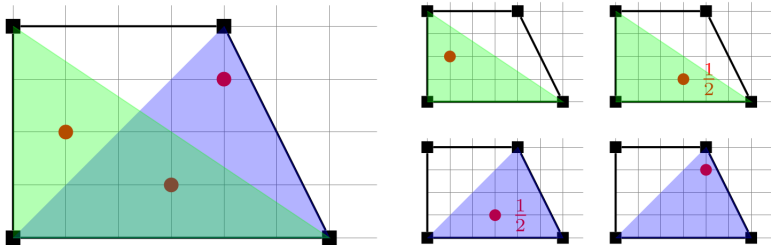
- **Input**  $f = \sum_{\alpha} b_{\alpha} X^{\alpha}$  of degree  $d$ ,  $\hat{\delta} \in \mathbb{Q}^{>0}$ ,  $\tilde{\delta} \in \mathbb{Q}^{>0}$   
Monomial squares = MoSq ( $f$ )    Complement = NoSq ( $f$ )

# 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 =  $\text{MoSq}(f)$     Complement =  $\text{NoSq}(f)$

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



# Algorithm optsonc: numerical steps

---

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

2 **Numerical** resolution of GEOMETRIC PROGRAM

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

# Algorithm optsonc: symbolic steps

---

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

GEOMETRIC PROGRAM provides “IN THEORY”

$$f_{\beta} = \sum_{\alpha} G_{\beta,\alpha} \cdot X^{\alpha} + b_{\beta} X^{\beta}, f + \sum_{\beta} G_{\beta,0} - b_0 = \sum_{\beta} f_{\beta} \geq 0$$

# Algorithm optsonc: symbolic steps

---

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

GEOMETRIC PROGRAM provides “IN THEORY”

$$f_{\beta} = \sum_{\alpha} G_{\beta,\alpha} \cdot X^{\alpha} + b_{\beta} X^{\beta}, \quad f + \sum_{\beta} G_{\beta,0} - b_0 = \sum_{\beta} f_{\beta} \geq 0$$

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”

$$f_{\beta} = \sum_{\alpha} G_{\beta,\alpha} \cdot X^{\alpha} + b_{\beta} X^{\beta}, f + \sum_{\beta} G_{\beta,0} - b_0 = \sum_{\beta} f_{\beta} \geq 0$$

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

- **Rounding step**  $\hat{G} \leftarrow \text{round}(\tilde{G}, \hat{\delta})$

# Algorithm optsonc: symbolic steps

---

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

GEOMETRIC PROGRAM provides “IN THEORY”

$$f_{\beta} = \sum_{\alpha} G_{\beta,\alpha} \cdot X^{\alpha} + b_{\beta} X^{\beta}, f + \sum_{\beta} G_{\beta,0} - b_0 = \sum_{\beta} f_{\beta} \geq 0$$

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

3 **Rounding step**  $\hat{G} \leftarrow \text{round}(\tilde{G}, \hat{\delta})$

4 **Projection step**

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

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

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



# Algorithm optsonc: symbolic steps

---

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

GEOMETRIC PROGRAM provides “IN THEORY”

$$f_{\beta} = \sum_{\alpha} G_{\beta,\alpha} \cdot X^{\alpha} + b_{\beta} X^{\beta}, f + \sum_{\beta} G_{\beta,0} - b_0 = \sum_{\beta} f_{\beta} \geq 0$$

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


3 **Rounding step**  $\hat{G} \leftarrow \text{round}(\tilde{G}, \hat{\delta})$

4 **Projection step**

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

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

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

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

# Algorithm optsage: numerical steps

---

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

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

# Algorithm optsage: numerical steps

---

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

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

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

$$\sum_j \mathbf{c}^{(j)} = \mathbf{b}, \quad \sum_i \alpha(i) \mathbf{v}_i^{(j)} = \mathbf{0}, \quad -\mathbf{1} \cdot \mathbf{v}_{\setminus j}^{(j)} = \mathbf{v}_j^{(j)}$$

$$\mathbf{c}_{\setminus j}^{(j)}, \mathbf{v}_{\setminus j}^{(j)} \geq \mathbf{0}, \quad D\left(\mathbf{v}_{\setminus j}^{(j)}, e\mathbf{c}_{\setminus j}^{(j)}\right) \leq \mathbf{c}_j^{(j)}$$

# Algorithm optsage: numerical steps

---

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

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

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

$$\sum_j \mathbf{c}^{(j)} = \mathbf{b}, \quad \sum_i \alpha(i) \mathbf{v}_i^{(j)} = \mathbf{0}, \quad -\mathbf{1} \cdot \mathbf{v}_{\setminus j}^{(j)} = v_j^{(j)}$$

$$\mathbf{c}_{\setminus j}^{(j)}, \mathbf{v}_{\setminus j}^{(j)} \geq \mathbf{0}, \quad D\left(\mathbf{v}_{\setminus j}^{(j)}, e \mathbf{c}_{\setminus j}^{(j)}\right) \leq c_j^{(j)}$$

**1 Numerical** resolution of RELATIVE ENTROPY PROGRAM

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

# Algorithm optsage: symbolic steps

---

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

# Algorithm optsage: symbolic steps

---

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

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

# Algorithm optsage: symbolic steps

---

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

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

2 **Rounding step**  $\hat{v} \leftarrow \text{round}(\tilde{v}, \hat{\delta}), \hat{c} \leftarrow \text{round}(\tilde{c}, \hat{\delta})$

# Algorithm opt sage: symbolic steps

---

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

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

- **2 Rounding step**  $\hat{v} \leftarrow \text{round}(\tilde{v}, \hat{\delta})$ ,  $\hat{c} \leftarrow \text{round}(\tilde{c}, \hat{\delta})$

- **3 Projection step**

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



# Algorithm opt sage: symbolic steps

---

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

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

- 2 **Rounding step**  $\hat{v} \leftarrow \text{round}(\tilde{v}, \hat{\delta})$ ,  $\hat{c} \leftarrow \text{round}(\tilde{c}, \hat{\delta})$

- 3 **Projection step**

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

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

$$c_j^{(j)} \geq D\left(v_{\setminus j}^{(j)}, ec_{\setminus j}^{(j)}\right) = \sum_{j \neq i > 1} v_i^{(j)} \log \frac{v_i^{(j)}}{ec_i^{(j)}} + v_1^{(j)} \log \frac{v_1^{(j)}}{ec_1^{(j)}}$$

# Algorithm opt sage: symbolic steps

---

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

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

- **2 Rounding step**  $\hat{v} \leftarrow \text{round}(\tilde{v}, \hat{\delta})$ ,  $\hat{c} \leftarrow \text{round}(\tilde{c}, \hat{\delta})$

- **3 Projection step**

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

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

$$c_j^{(j)} \geq D\left(v_{\setminus j}^{(j)}, ec_{\setminus j}^{(j)}\right) = \sum_{j \neq i > 1} v_i^{(j)} \log \frac{v_i^{(j)}}{ec_i^{(j)}} + v_1^{(j)} \log \frac{v_1^{(j)}}{ec_1^{(j)}}$$

$$c_1^{(j)} \leftarrow \text{round} \uparrow (\exp(\dots), \hat{\delta})$$

# Algorithm opt sage: symbolic steps

---

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

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

- **2 Rounding step**  $\hat{v} \leftarrow \text{round}(\tilde{v}, \hat{\delta})$ ,  $\hat{c} \leftarrow \text{round}(\tilde{c}, \hat{\delta})$


- **3 Projection step**

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

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

$$c_j^{(j)} \geq D(v_{\setminus j}^{(j)}, ec_{\setminus j}^{(j)}) = \sum_{j \neq i > 1} v_i^{(j)} \log \frac{v_i^{(j)}}{ec_i^{(j)}} + v_1^{(j)} \log \frac{v_1^{(j)}}{ec_1^{(j)}}$$

$$c_1^{(j)} \leftarrow \text{round} \uparrow (\exp(\dots), \hat{\delta})$$

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

# SOS Benchmarks

---

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

Id	$n$	$d$	RealCertify		RoundProject		RAGLib	CAD
			$\tau_1$ (bits)	$t_1$ (s)	$\tau_2$ (bits)	$t_2$ (s)	$t_3$ (s)	$t_4$ (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

---

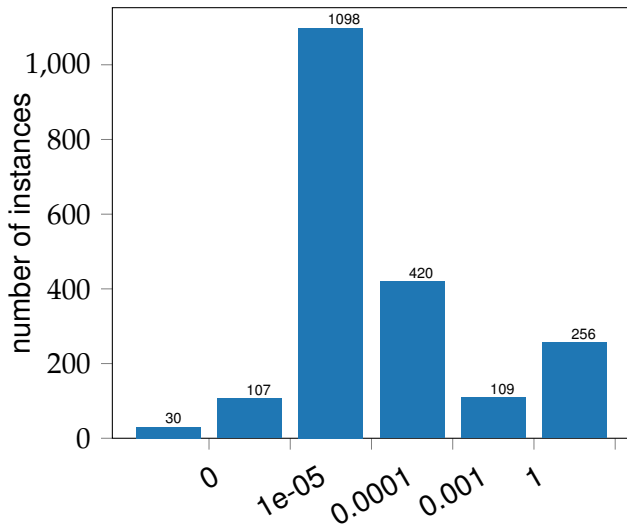
terms	bit size		time	
	optsonc	optsage	optsonc	optsage
6	432	1005	0.06	0.26
9	806	2696	0.19	0.66
12	1261	5568	0.37	1.29
20	2592	19203	0.64	4.00
24	3826	32543	0.97	6.66
30	5029	53160	1.34	10.58
50	10622	167971	3.95	32.78

💡 “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  $\tau$

Algo	Input	$\mathbf{K}$	OUTPUT BIT SIZE
intsos	$\overset{\circ}{\Sigma}$	$\mathbb{R}^n$	$\tau d^{d^{\mathcal{O}(n)}}$
intsage	$\overset{\circ}{C}_{\text{SAGE}}$	$\mathbb{R}^n$	

# Conclusion and Perspectives

---

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

Algo	Input	$\mathbf{K}$	OUTPUT BIT SIZE
intsos	$\overset{\circ}{\Sigma}$	$\mathbb{R}^n$	$\tau d^{d^{\mathcal{O}(n)}}$
intsage	$\overset{\circ}{C}_{\text{SAGE}}$	$\mathbb{R}^n$	

- 💡 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, *ISSAC'19*. arxiv:1902.02123