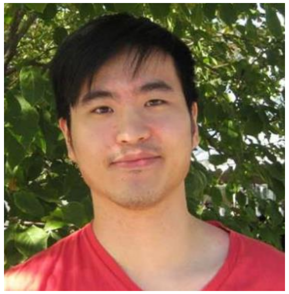
Complexity of Finding Local Minima in Polynomial Optimization

(and a little bit on Higher-Order Newton Methods)

Amir Ali Ahmadi
Princeton University

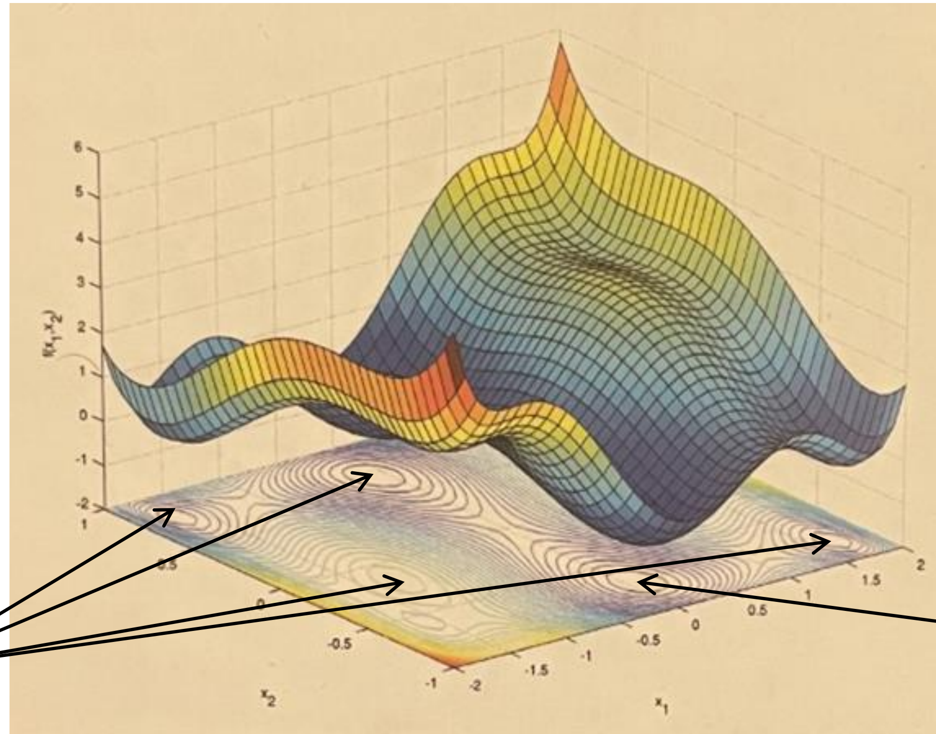
Joint work with:



Jeffrey Zhang
Yale University

A natural question in nonconvex optimization

local min



global min

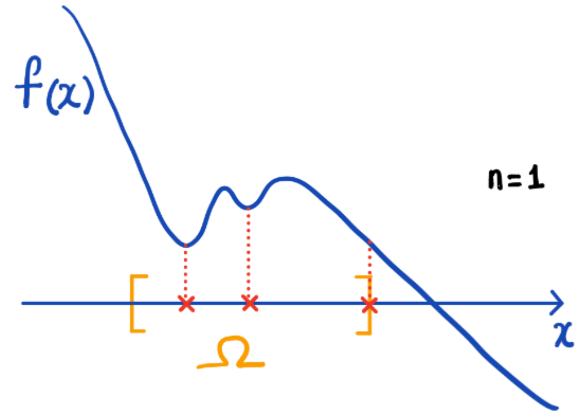
If global minima are hard to reach, can we at least find a local min efficiently?

Local minima

$$\min_{x \in \mathbb{R}^n} \; f(x) \quad \text{s.t.} \quad x \in \Omega$$

$\bar{x} \in \Omega$ is a local minimum if

$$\exists \, \delta > 0 \;\; \text{s.t.} \;\; \left.\begin{array}{l} \|x - \bar{x}\| \leq \delta \\ x \in \Omega \end{array}\right\} \Rightarrow f(\bar{x}) \leq f(x)$$



Let's focus on the following setting:

$$\Omega = \{ x \in \mathbb{R}^n \mid g_i(x) \geq 0, \; i = 1, \ldots, m \}$$

where $f, g_1, \ldots, g_m$ are polynomials of degree $\leq d$

e.g.

$$\min. \;\; 5 x_1 x_2^2 - \frac{3}{4} x_2$$

$$\text{s.t.} \;\; \frac{1}{2} x_1 + 1 x_2 \geq 1$$

$$-3 x_1^2 - 2 x_2^2 \geq \frac{9}{2}$$

problem input

Simplest cases to consider:

1. **Ω polyhedral, *f* linear (i.e., linear programming)**
   - local minima=global minima
   - can be found in polynomial time [Khachiyan], [Karmarkar]
2. **Ω polyhedral, *f* quadratic (i.e., quadratic programming)**
   - global minima NP-hard to find [Pardalos, Vavasis]
   - what about local minima? **(first focus of this talk)**

Finding a local minimizer of a quadratic program

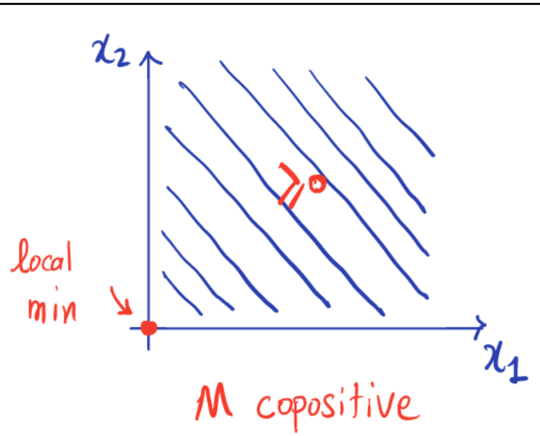
Open questions in complexity theory for numerical optimization

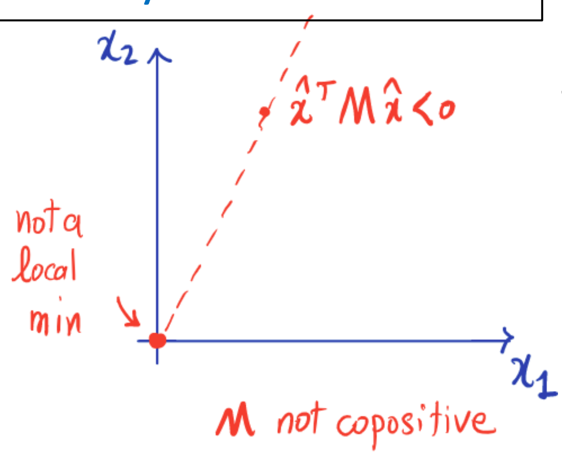
Panos M. Pardalos and Stephen A. Vavasis

May 5, 1992

Problem 3. What is the complexity of finding even a local minimizer for nonconvex quadratic programming, assuming the feasible set is compact? Murty and Kabadi (1987) and Pardalos and Schnitger (1988) have shown that it is NP-hard to test whether a given point for such a problem is a local minimizer, but that does not rule out the possibility that another point can be found that is easily verified as a local minimizer.

- A matrix M is copositive if $x^T M x \geq 0, \forall x \geq 0$
- NP-hard to check copositivity





However,

$x^T M x$ has a local min over $x \geq 0$ $\not\Rightarrow$ M copositive

e.g., $M = \begin{pmatrix} 0 & 1 \\ 1 & -2 \end{pmatrix}$ → $x^T M x$ has local min at $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$; → not copositive

Finding a local minimizer of a quadratic program

Open questions in complexity theory for numerical optimization

Panos M. Pardalos and Stephen A. Vavasis

May 5, 1992

Problem 3. What is the complexity of finding even a local minimizer for nonconvex quadratic programming, assuming the feasible set is compact? Murty and Kabadi (1987) and Pardalos and Schnitger (1988) have shown that it is NP-hard to test whether a given point for such a problem is a local minimizer, but that does not rule out the possibility that another point can be found that is easily verified as a local minimizer.

Remarks:
- when Ω is compact, existence of a local min is guaranteed
- there is always a rational local min with polynomial bitsize [Vavasis]

Theorem [AAA, Zhang]

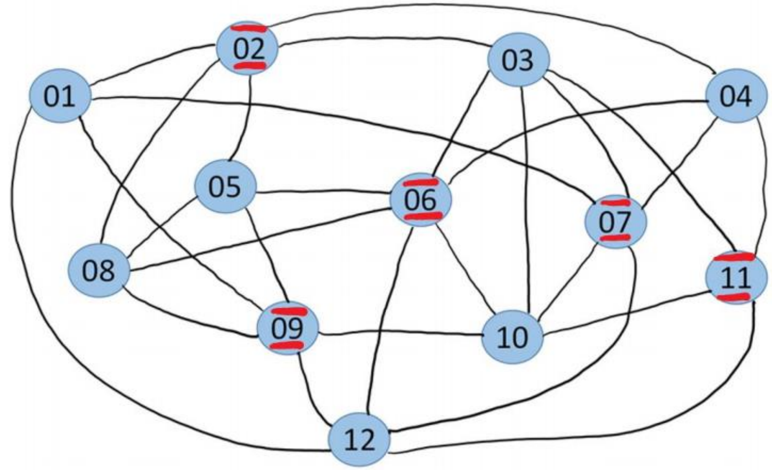
Unless P = NP, there cannot be a polynomial-time algorithm that finds a point within Euclidean distance c^n (for any constant $c \geq 0$) of a local minimum of an n-variate quadratic polynomial over a polytope.

To prove this, let's move from continuous to discrete…

Stable sets in graphs

A **stable set** in a graph G is a subset of the vertices of G that are pairwise non-adjacent

The size of the largest such set is the **stability number** of G, denoted by $\boldsymbol{\alpha(G)}$



$\boldsymbol{\alpha(G) = 5}$

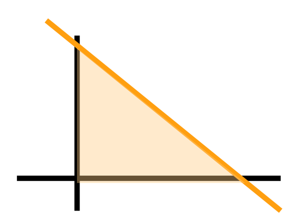
Given a graph G on n nodes and an integer $r \in \{1, \ldots, n\}$, it is **NP-hard to decide if $\boldsymbol{\alpha(G) \geq r}$** [Karp]

We show: a poly-time algorithm that gets within distance c^n of a local min of the following program, would decide if $\alpha(G) \geq r$ in poly time.

$$\min_{x \in \mathbb{R}^n} \quad x^T\left(\left(r - \tfrac{1}{2}\right)(A + I) - J\right)x$$

$$\text{s.t.} \quad x \geq 0$$

$$\sum_{i=1}^{n} x_i \leq 3\lceil c^n \sqrt{n} \rceil$$



A: adjacency matrix of G

I: identity matrix

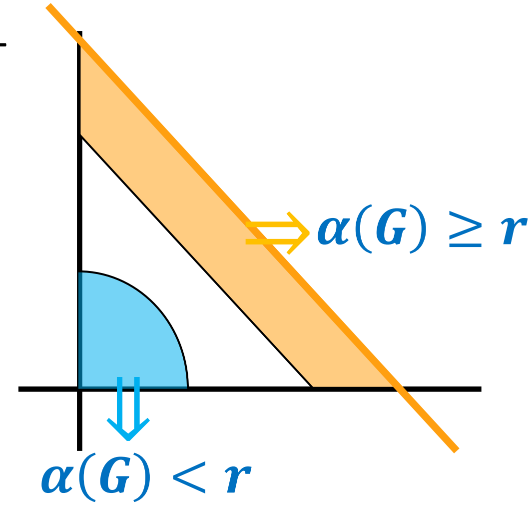
J: all-ones matrix

Proof outline

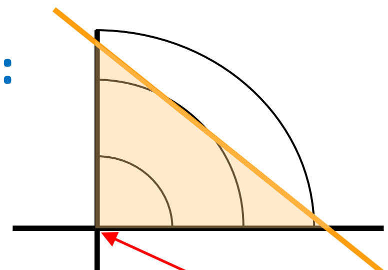
$$\min_{x \in \mathbb{R}^n} \; x^T\left(\left(r-\tfrac{1}{2}\right)(A+I)-J\right)x$$
$$\text{s.t.} \quad x \geq 0$$
$$\sum_{i=1}^{n} x_i \leq 3\lceil c^n \sqrt{n} \rceil$$

- Suppose there was a poly-time algorithm that given a quadratic program, landed within distance c^n of one of its local minima

- Run this algorithm on the constructed instances



$\alpha(G) < r$:



Only local minimum is here

$\alpha(G) \geq r$:

All local minima are here

Implication (e.g., via [Ogiwara, Watanabe], [Mahaney]):
- Pick any poly-time algorithm that attempts to find a local min of a QP.

- Unless P=NP, out of the QP instances that can be encoded with k bits, **this algorithm will land far away from any local min** on a number of instances that is larger than any polynomial in k.

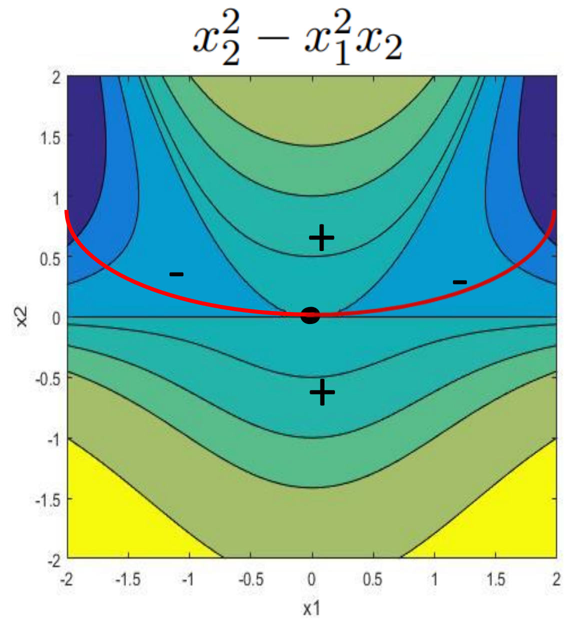
For details, see arxiv/2008.05558

Finding local minima in the *unconstrained* case

Given a polynomial of degree d, can we test if it has a local minimum (and if so, can we find one efficiently)?

- **When $d \geq 4$,** strongly NP-hard [AAA, Zhang] (key to the proof of the previous result)
- **When $d = 2$**, poly-time solvable
  - Find a critical point (linear system)
  - Check if the Hessian is positive semidefinite
- **What about $d = 3$?**

Unlike quadratic functions,
- First and second order conditions do not characterize local minima
- Lack of a descent direction does not imply local minimality



To start, can even **recognize** a local min of a cubic polynomial efficiently?

A characterization of local minima for cubics

Theorem (AAA, Zhang) ["third-order condition"]

A point $z \in \mathbb{R}^n$ *is a local min of a cubic polynomial* $p: \mathbb{R}^n \to \mathbb{R}$ *if and only if*

- $\nabla p(z) = 0, \nabla^2 p(z) \succcurlyeq 0,$ *(FONC, SONC)*
- $d \in Null(\nabla^2 p(z)) \Rightarrow \nabla p_3(d) = 0. (TOC)*$

Moreover, these conditions can be checked in polynomial time.

* p_3 here denotes the cubic homogenous component of p.

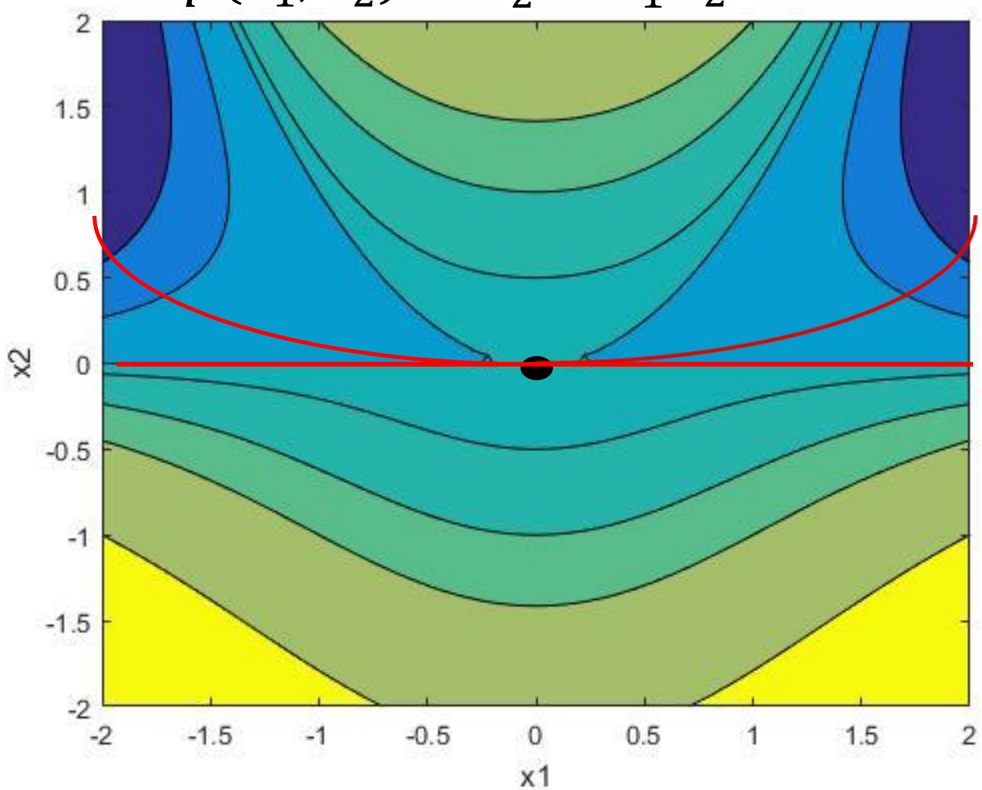
Some intuition on third-order condition (TOC)

$$d \in Null(\nabla^2 p(\bar{x})) = 0 \Rightarrow p_3(d) = 0$$

$\nabla p_3(d) = 0$ ⇗ ↘ Euler's identity:

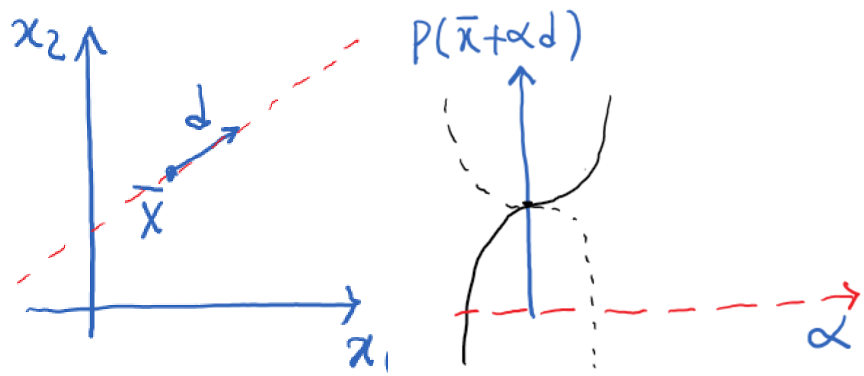
$$p_3(d) = \frac{1}{3} d^T \nabla p_3(d)$$

$$p(x_1, x_2) = x_2^2 - x_1^2 x_2$$



Necessary for C^3 functions (where p_3 would be the cubic component of the Taylor expansion).

"Third Order Necessary Condition" (TONC)



Not sufficient for local optimality, even for cubics

Guarantees no descent directions for cubic polynomials

Does not guarantee no parabolas of descent

Recognizing local minima of cubics in poly time

- Input: $p, \bar{x}$
- Compute gradient and Hessian of p at $\bar{x}$
- Check FONC and SONC

- Compute a basis $\{v_1, v_2, \ldots, v_k\}$ for null space of $\nabla^2 p(\bar{x})$ in poly time
  - E.g., by solving a sequence of linear systems

- Compute gradient of p_3, evaluated on the null space of $\nabla^2 p(\bar{x})$

$$\nabla p_3(x) = \begin{pmatrix} \frac{\partial p_3}{\partial x_1}(x) \\ \cdots \\ \frac{\partial p_3}{\partial x_n}(x) \end{pmatrix} \Rightarrow \begin{pmatrix} \frac{\partial p_3}{\partial x_1}(\alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_k v_k) \\ \cdots \\ \frac{\partial p_3}{\partial x_n}(\alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_k v_k) \end{pmatrix} \longleftarrow \begin{pmatrix} g_1(\alpha_1, \ldots, a_k) \\ \cdots \\ g_n(\alpha_1, \ldots, a_k) \end{pmatrix}$$

- All coefficients of all g_i must be zero

Back to the existence/search question

- So we can check if a given point is a local min efficiently.
- But can we *find* a local min of a cubic polynomial efficiently?

Let's start with a "simpler" question. Can we find a critical point efficiently?

Unfortunately…

Theorem (AAA, Zhang)

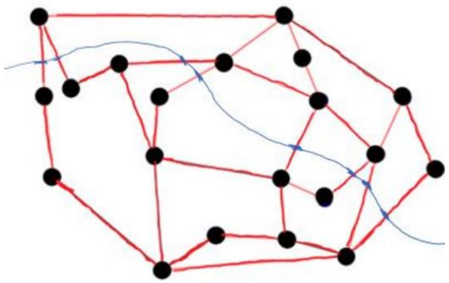
Deciding if a cubic polynomial has a critical point is strongly NP-hard.

But somewhat surprisingly:

Theorem (AAA, Zhang)

A local minimum of a cubic polynomial can be found by solving linearly many semidefinite programs of size linear in the number of variables.

NP-hardness of testing if a cubic has a critical point



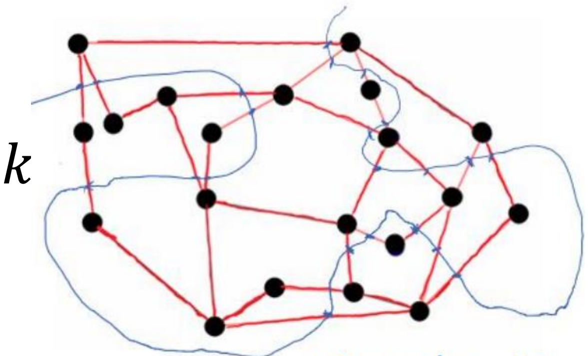
▪Cut value=8

MAXCUT

Does the graph have a cut of size k

Quadratic satisfiability

$$\frac{1}{4}\sum_{(i,j)\in E}(1-x_i x_j)=k \qquad 1-x_i^2=0, i=1,\dots,n$$



▪Cut value=23 (optimal)

Critical points of a cubic polynomial

$$p(x_1,\dots,x_n,y_0,y_1,\dots,y_n)=y_o\left(\frac{1}{4}\sum_{(i,j)\in E}(1-x_i x_j)-k\right)+\sum_{i=1}^{n}y_i(1-x_i^2)$$

NP-hardness of testing if a cubic has a critical point

$$p(x,y) = y_o\left(\frac{1}{4}\sum_{(i,j)\in E}(1 - x_i x_j) - k\right) + \sum_{i=1}^{n} y_i(1 - x_i^2)$$

$$\nabla p(x,y) = \begin{bmatrix} \frac{dp}{dx_i} \\ \frac{dp}{dy_0} \\ \frac{dp}{dy_i} \end{bmatrix} = \begin{bmatrix} -\frac{y_0}{4}\left(\sum_{(i,j)\in E} x_j\right) - 2x_i y_i \\ \frac{1}{4}\sum_{(i,j)\in E}(1 - x_i x_j) - k \\ 1 - x_i^2 \end{bmatrix}$$

Any cut of size k $\Rightarrow$ critical point $(x = \text{cut}, y = 0)$

Any critical point $\Rightarrow$ cut of size k $(x \Rightarrow \text{cut})$

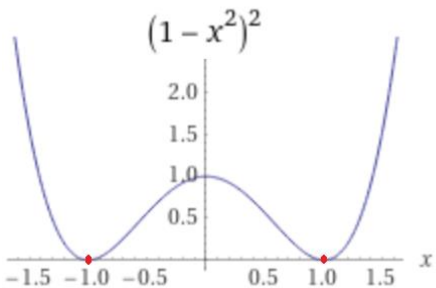
But this doesn't necessarily mean finding local minima is NP-hard.
Let's understand the geometry better…

Unexpected convexity

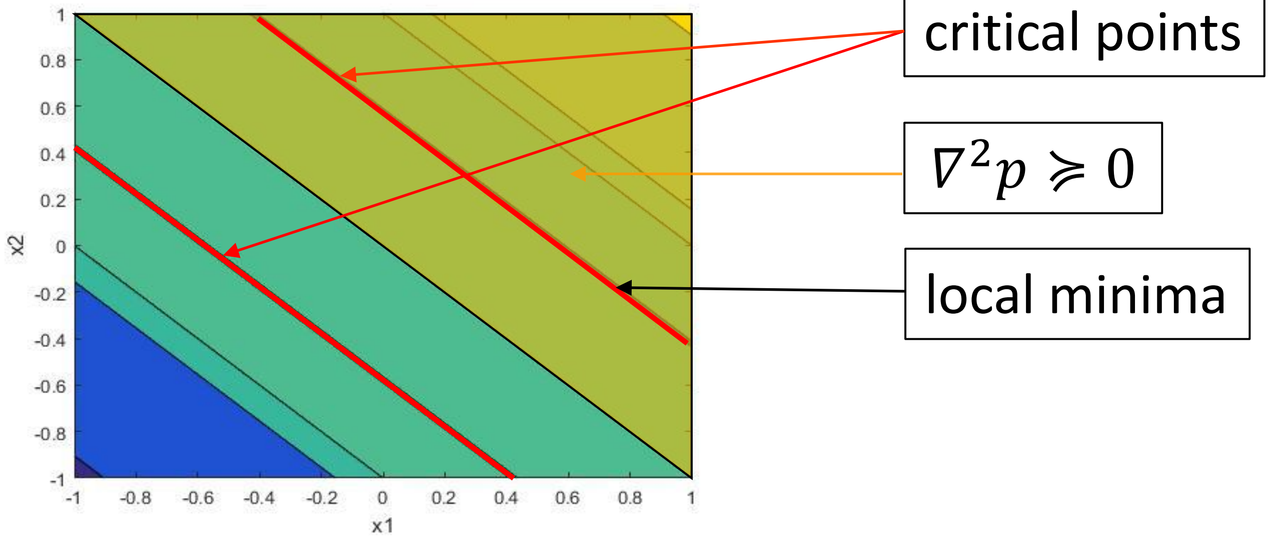
Theorem (AAA, Zhang)

The set of local minima of any cubic polynomial is ***convex****.*

Not true for quartics:



$$p(x_1, x_2) = x_1^3 + 3x_1^2x_2 + 3x_1x_2^2 + x_2^3 - x_1 - x_2$$



Proof based on our previous characterization:

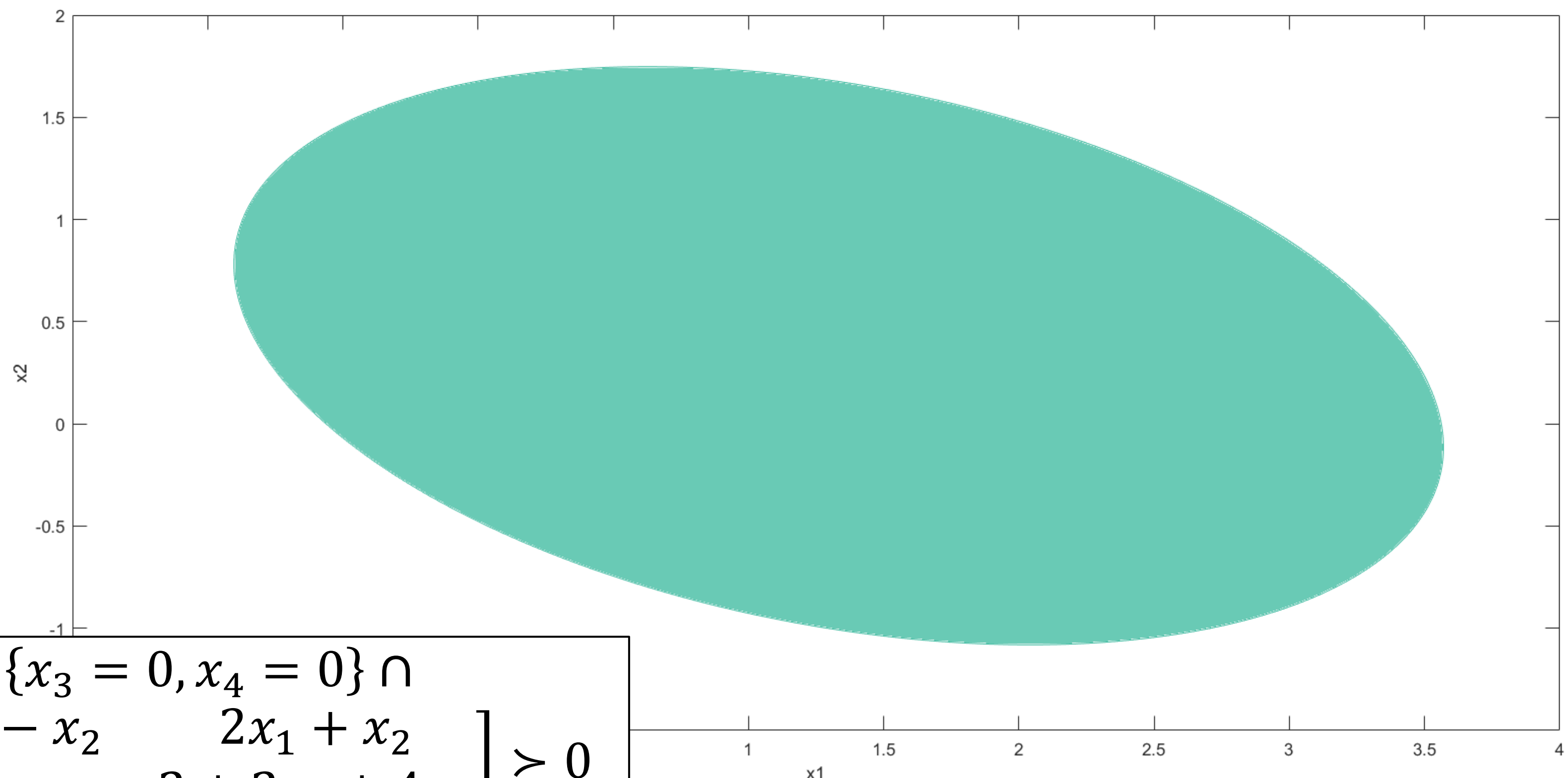
Theorem (AAA, Zhang) ["third-order condition"]

A point $z \in \mathbb{R}^n$ is a local min of a cubic polynomial p if and only if

- $\nabla p(z) = 0, \nabla^2 p(z) \succcurlyeq 0,$
- $d \in Null\left(\nabla^2 p(z)\right) \Rightarrow \nabla p_3(d) = 0.$

Set of local minima not necessarily polyhedral

$$p(x_1, x_2, x_3, x_4) = \frac{1}{2}x_1^2x_3^2 + 2x_1x_3x_4 + \frac{1}{2}x_1x_4^2 - \frac{1}{2}x_2x_3^2$$
$$+x_2x_3x_4 + 2x_2x_4^2 + x_3^2 + x_4^2$$

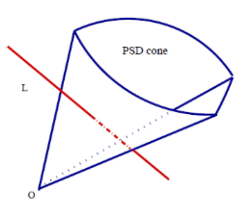


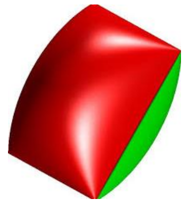
$$\{x_3 = 0, x_4 = 0\} \cap \begin{bmatrix} 2 + x_1 - x_2 & 2x_1 + x_2 \\ 2x_1 + x_2 & 2 + 2x_1 + 4x_2 \end{bmatrix} \succ 0$$

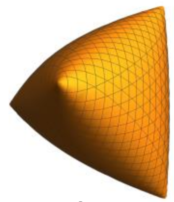
Local minima of cubics and SDP

Theorem (AAA, Zhang)

The set of local minima of any cubic polynomial is ***semidefinite representable****.*







(though not always polyhedral)

In fact, one can show a converse statement:

Theorem (AAA, Zhang)

The interior of any semidefinite representable set is the projection of the set of local minima of some cubic polynomial.

⇒ Any algorithm for finding local minima of cubic polynomials can be turned into an SDP solver

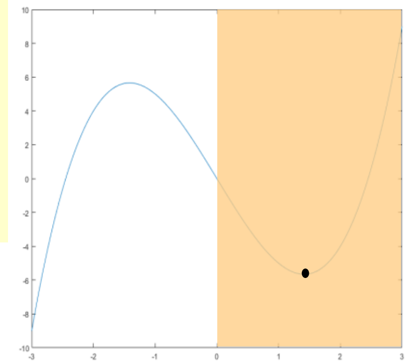
Informally: **"Complexity of SDP = Complexity of local minima of cubics"**

Connection to "sums of squares" (SOS)

Theorem (AAA, Zhang)

If a cubic polynomial p has a local minimum, then the closure of the set of its local minima equals the solution set of:

$$\min_{x}.\ p(x) \quad \text{s.t. } \nabla^2 p(x) \succcurlyeq 0$$



$$\min_{x} p(x) \ \text{s.t. } \nabla^2 p(x) \succcurlyeq 0 \quad \geq \quad \max_{\sigma(x), S(x), \gamma} \gamma \quad \text{s.t. } p(x) - \gamma = \sigma(x) + Tr\left(\nabla^2 p(x) S(x)\right), \forall x$$

σ is SOS ← $\sigma(x) = \sum q_i^2(x)$

S is an SOS-matrix ← $S(x) = R(x)R(x)^T$

(this is an SDP in disguise)

Theorem (AAA, Zhang)

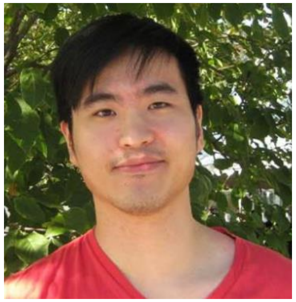
If p has a second-order point, the first level of this SOS relaxation (i.e., when $\deg(\sigma) = \deg(S) = 2$) is tight.

PRINCETON UNIVERSITY ORFE

For details, see arxiv/2008.06148

Higher-Order Newton Methods with Polynomial Work per Iteration

Joint work with:



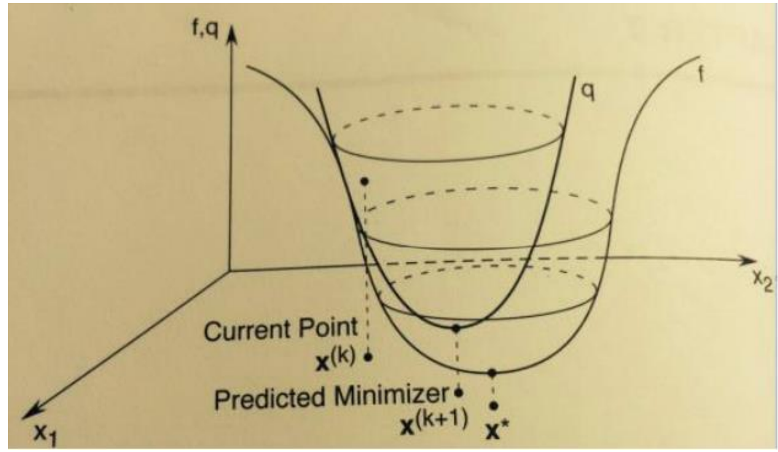
Abraar Chaudhry
Princeton University

Jeffrey Zhang
Yale University

The Newton method

- **Newton's algorithm:**
  2nd order Taylor expand, move to minimizer; iterate.

- Work per iteration: Solving a linear system (poly-time in dimension)



- **Classical theorem in optimization:**
  Newton's method has **local quadratic convergence**

$$||x_{k+1} - x^*|| \leq c||x_k - x^*||^2$$

Newton

k	x_k
1.000000000000000	12.00000000000000
2.000000000000000	8.023148148148149
3.000000000000000	5.400548660419450
4.000000000000000	3.714654390828676
5.000000000000000	2.718005659267038
6.000000000000000	2.263213061967483
7.000000000000000	2.159579216386407
8.000000000000000	2.154446935535738
9.000000000000000	2.154434690101485
10.000000000000000	2.154434690031884
11.000000000000000	2.154434690031884

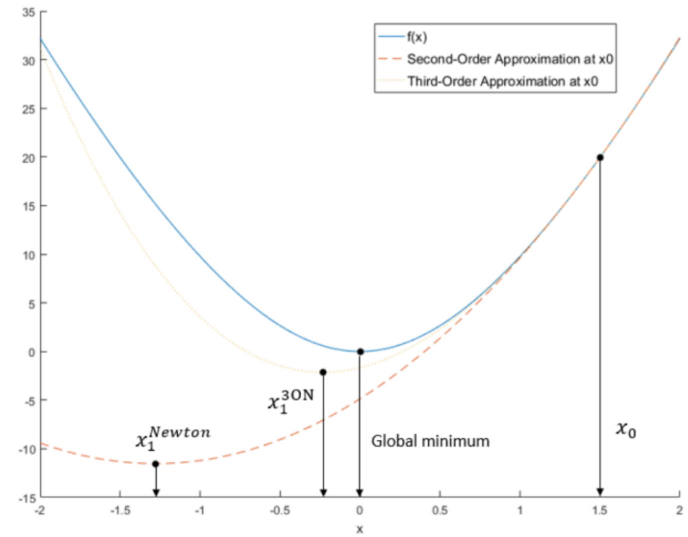
Circled in red: correct significant digits

Q: Why not Taylor expand to higher order?!

- Because minimizing multivariate polynomials of higher degree is NP-hard

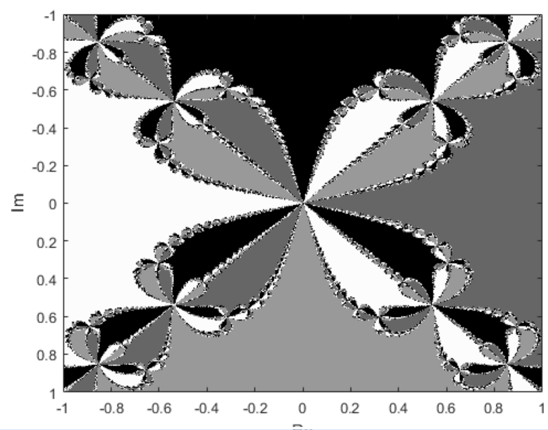
An unregularized third-order Newton method

- **Newton's method:** 2^{nd} order Taylor expand, move to minimizer
- **3^{rd}-order Newton:** 3^{rd} order Taylor expand, move to local minimizer (by SDP)
  - Polynomial iteration complexity

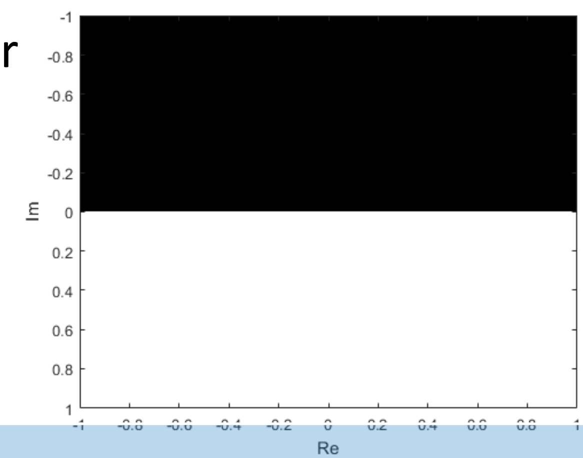


Sensitivity to initialization:

Newton



3^{rd}-order Newton



Theorem (informal) [Silina, Zhang]

Under standard assumptions, the 3^{rd}-order unregularized Newton method has local ***cubic convergence*** *(i.e., $||x_{k+1} - x^*|| \leq c||x_k - x^*||^3$).*

Higher-order Newton with polynomial work per iteration

Let $f: \mathbb{R}^n \to \mathbb{R}$ be the function we wish to minimize.

Our algorithm ("d^{th} degree Newton") :

1. Taylor expand f to order d around the current iterate x_k
   Denote the resulting degree-d polynomial by T_{d,x_k}

2. Find the "closest" *sos-convex* polynomial of degree $d+1$ (if d is odd) or $d+2$ (if d is even) to T_{d,x_k}
   Denote the resulting (sos-convex) polynomial by ψ_{d,x_k}

3. Let x_{k+1} be a minimizer of ψ_{d,x_k}

Claim: Steps 2 and 3 take polynomial time (in dimension): they reduce to SDPs

Theorem: The above algorithm has **local convergence of order $\boldsymbol{d}$**:

$$\left|\left|\boldsymbol{x_{k+1}} - \boldsymbol{x}^*\right|\right| \leq \boldsymbol{c}\left|\left|\boldsymbol{x_k} - \boldsymbol{x}^*\right|\right|^{\boldsymbol{d}}$$

(leads to lower oracle complexity compared to Newton)

Basin of attraction

Newton in dimension one:

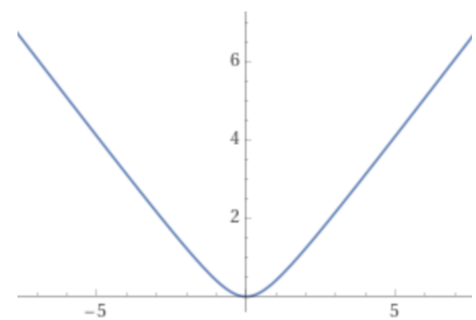
$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

Our method in dimension one and with $d = 3$:

$$x_{k+1} = x_k - 2\frac{f''(x_k)}{f'''(x_k)} - \sqrt[3]{\frac{12 f'(x_k) f''(x_k)}{\left(f'''(x_k)\right)^3} - \left(2\frac{f''(x_k)}{f'''(x_k)}\right)^3}$$

Example. Minimize:

$$f(x) = \sqrt{x^2 + 1} - 1$$



Newton's basin: $|x_0| < 1$

Our third-order method's basin: $|x_0| < 3.407$

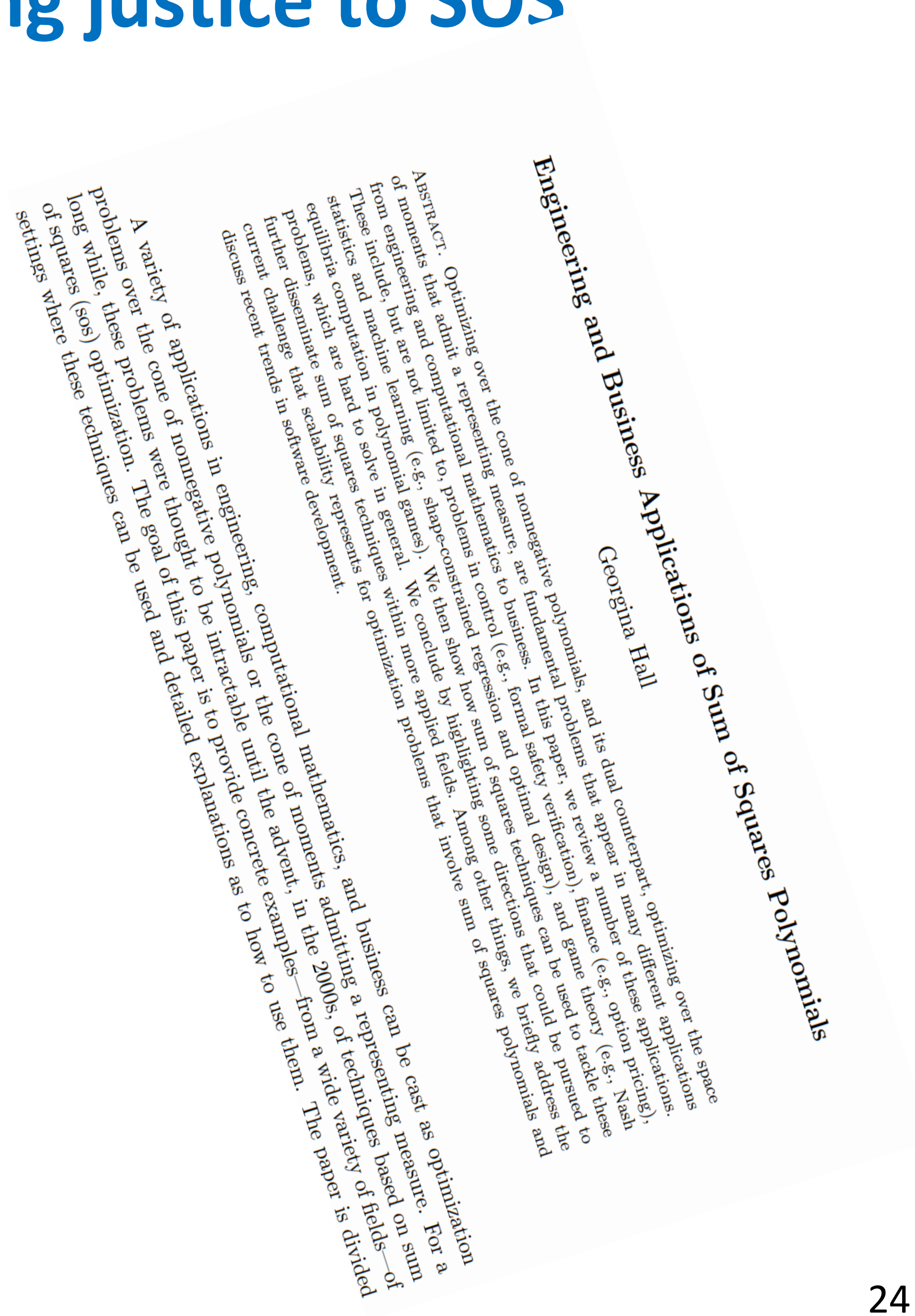
(exact cutoff: $\sqrt{\frac{1}{3}\left(11 + \frac{142}{\sqrt[3]{1691+9i\sqrt{47}}} + \sqrt[3]{1691 + 9i\sqrt{47}}\right)}$)

Degree	Radius of Convergence
2 (Regular Newton)	1
3	∼3.4
4	∼4.5
5	∼5.9

This talk not doing justice to SOS

- Sum of squares optimization is a powerful machinery to *globally* solve any (nonconvex) optimization problem modelled by polynomial equations and inequalities

- Nice survey by Georgina Hall (INSEAD)
    - Covers applications in finance, game theory, statistics and ML, control, etc.

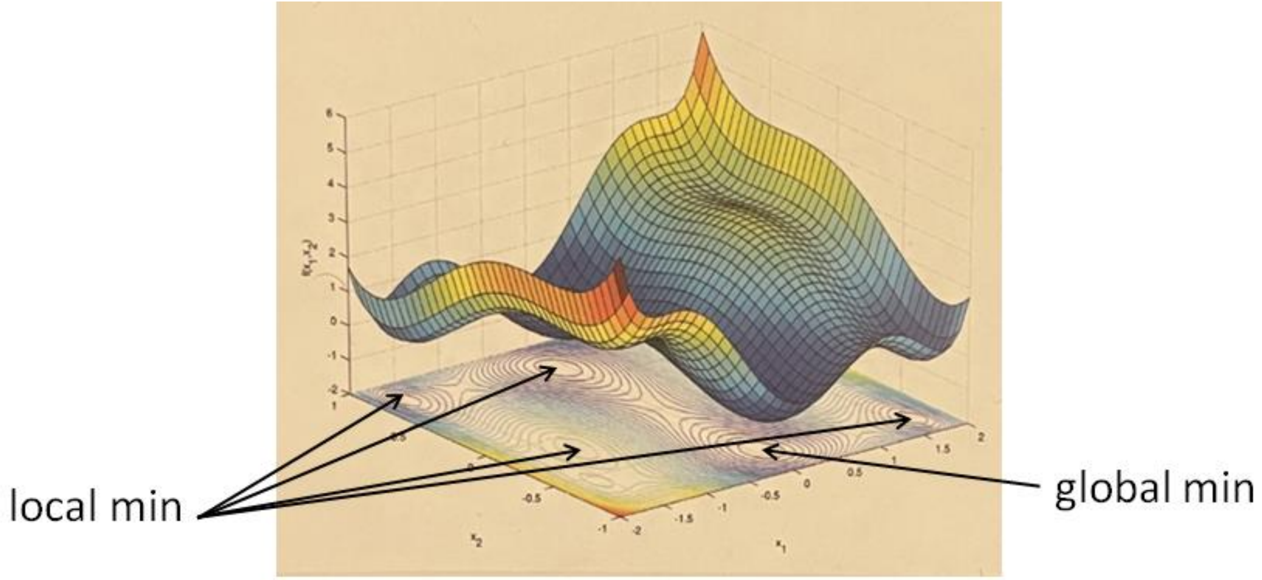

Engineering and Business Applications of Sum of Squares Polynomials

Georgina Hall

ABSTRACT. Optimizing over the cone of nonnegative polynomials, and its dual counterpart, optimizing over the space of moments that admit a representing measure, are fundamental problems that appear in many different applications from engineering and computational mathematics to business. In this paper, we review a number of these applications. These include, but are not limited to, problems in control (e.g., formal safety verification), finance (e.g., option pricing), statistics and machine learning (e.g., shape-constrained regression and optimal design), and game theory (e.g., Nash equilibria computation in polynomial games). We then show how sum of squares techniques can be used to tackle these problems, which are hard to solve in general. We conclude by highlighting some directions that could be pursued to further disseminate sum of squares techniques within more applied fields. Among other things, we briefly address the current challenge that scalability represents for optimization problems that involve sum of squares polynomials and discuss recent trends in software development.

A variety of applications in engineering, computational mathematics, and business can be cast as optimization problems over the cone of nonnegative polynomials or the cone of moments admitting a representing measure. For a long while, these problems were thought to be intractable until the advent, in the 2000s, of techniques based on sum of squares (sos) optimization. The goal of this paper is to provide concrete examples—from a wide variety of fields—of settings where these techniques can be used and detailed explanations as to how to use them. The paper is divided

One takeaway message



Want to know more?

- Finding local minima for QPs:
Mathematical Programming, 2022
arxiv/2008.05558

- Finding local minima for cubics:
Advances in Mathematics, 2022
arxiv/2008.06148

aaa.princeton.edu
sites.google.com/view/jeffreyzhang
chaudhrya.github.io