Deciding Non-linear Numerical Constraints: an Overview

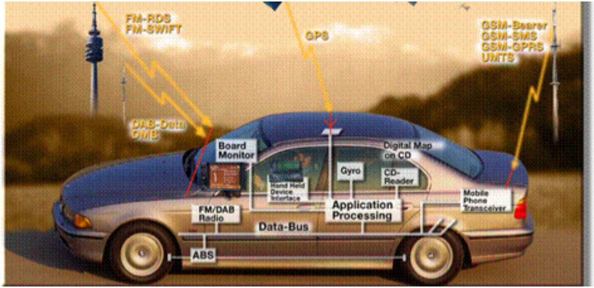
Stefan Ratschan

Academy of Sciences of the Czech Republic

January 30, 2010

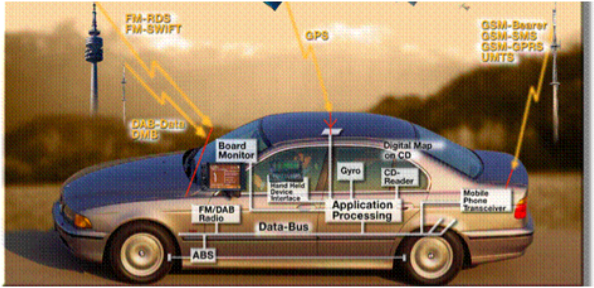
Motivation I

By far most micro-processors nowadays do not occur in desktop PC's but embedded in technical systems (trains, cars, robots, your washing machine etc.)



Motivation I

By far most micro-processors nowadays do not occur in desktop PC's but embedded in technical systems (trains, cars, robots, your washing machine etc.)



Models of technical systems usually in numerical domains.

Motivation II

Continuous is simpler then discrete!

Motivation II

Continuous is simpler then discrete!

	integers	reals
sat. of linear constraints	NP-hard	polynomial time
sat. of polynomial constraints	undecidable	decidable

Motivation II

Continuous is simpler then discrete!

	integers	reals
sat. of linear constraints	NP-hard	polynomial time
sat. of polynomial constraints	undecidable	decidable

So: to solve discrete problem,
exploit corresponding continuous problem ("relaxation").

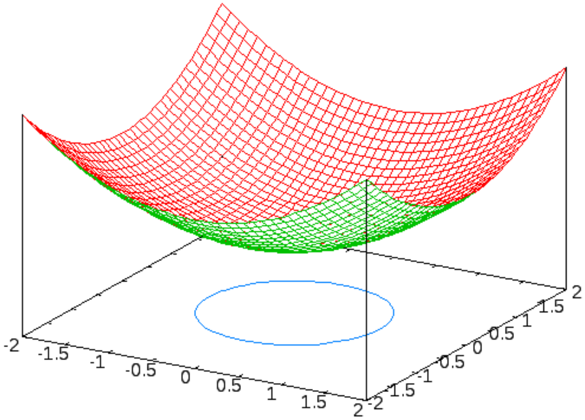
Example: MILP

Example

$$x^2 + y^2 - 1 = 0 \wedge y - x^2 = 0$$

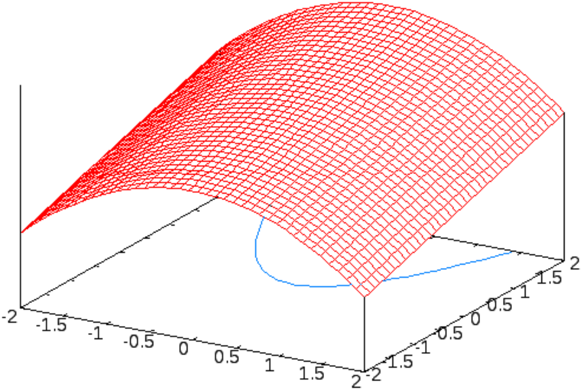
Example

$$x^2 + y^2 - 1 = 0 \wedge y - x^2 = 0$$



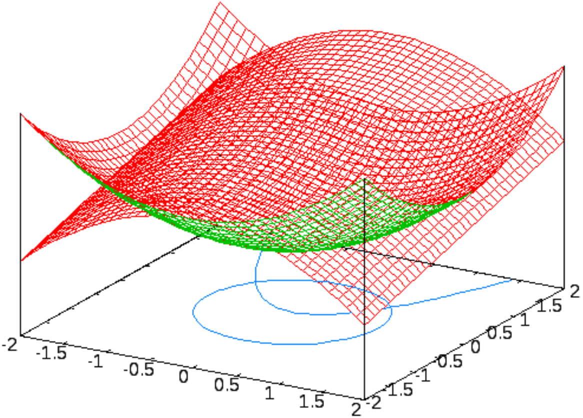
Example

$$x^2 + y^2 - 1 = 0 \wedge y - x^2 = 0$$

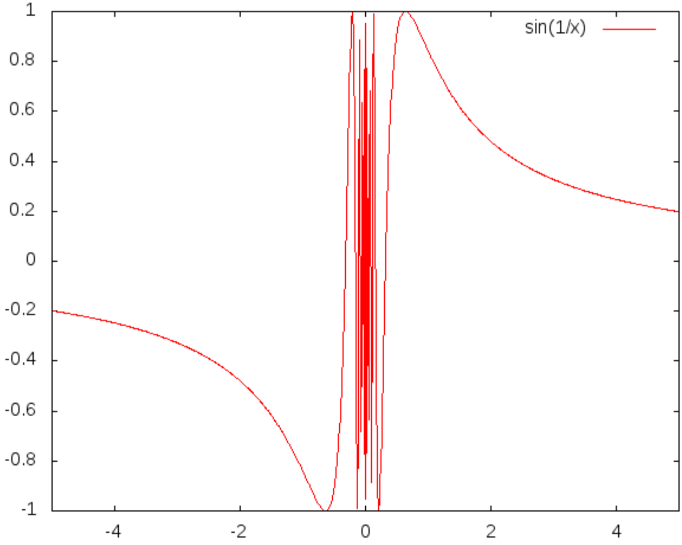


Example

$$x^2 + y^2 - 1 = 0 \wedge y - x^2 = 0$$



Example



Problem Definition

Given: formula in certain sub-class of $FO(\mathbb{R}, =, \leq, <, +, \times, \text{sin}, \dots)$

Decide: `sat/unsat`

Problem Definition

Given: formula in certain sub-class of $FO(\mathbb{R}, =, \leq, <, +, \times, \sin, \ldots)$

Decide: $\texttt{sat}/\texttt{unsat}$ + certificate (if possible)

Problem Definition

Given: formula in certain sub-class of $\mathit{FO}(\mathbb{R}, =, \leq, <, +, \times, \sin, \ldots)$

Decide: $\texttt{sat}/\texttt{unsat}$ + certificate (if possible)

Subclasses: quantifier-free, polynomial, linear, ...

Contents

Certificates for Satisfiability

Quantifier-free case: (e.g., $x^2 = 2$)

Certificates for Satisfiability

Quantifier-free case: (e.g., $x^2 = 2$)

Certificate: satisfying valuation (*solution*)

Certificates for Satisfiability

Quantifier-free case: (e.g., $x^2 = 2$)

Certificate: satisfying valuation (*solution*)

But: how to represent solution?

Certificates for Satisfiability

Quantifier-free case: (e.g., $x^2 = 2$)

Certificate: satisfying valuation (*solution*)

But: how to represent solution?

Linear case: rational number (e.g., $3x = 2 \quad x \mapsto \frac{2}{3}$)

Certificates for Satisfiability

Quantifier-free case: (e.g., $x^2 = 2$)

Certificate: satisfying valuation (*solution*)

But: how to represent solution?

Linear case: rational number (e.g., $3x = 2 \quad x \mapsto \frac{2}{3}$)

Polynomial case:

- in general, no expression in terms of roots (Abel-Ruffini theorem),
- real algebraic numbers (unintuitive, inefficient [Roy and Szpirglas, 1990]), $x < y$?

Certificates for Satisfiability

Quantifier-free case: (e.g., $x^2 = 2$)

Certificate: satisfying valuation (*solution*)

But: how to represent solution?

Linear case: rational number (e.g., $3x = 2 \quad x \mapsto \frac{2}{3}$)

Polynomial case:

- in general, no expression in terms of roots (Abel-Ruffini theorem),
- real algebraic numbers (unintuitive, inefficient [Roy and Szpirglas, 1990]), $x < y$?

In general: (arbitrary precise) approximation

Certificates for Satisfiability

Quantifier-free case: (e.g., $x^2 = 2$)

Certificate: satisfying valuation (*solution*)

But: how to represent solution?

Linear case: rational number (e.g., $3x = 2 \quad x \mapsto \frac{2}{3}$)

Polynomial case:

- in general, no expression in terms of roots (Abel-Ruffini theorem),
- real algebraic numbers (unintuitive, inefficient [Roy and Szpirglas, 1990]), $x < y$?

In general: (arbitrary precise) approximation $5 \mapsto 4.6557\ldots$

Certificates for Unsatisfiability

Example: $p(x) < 0$

Certificates for Unsatisfiability

Example: $p(x) < 0$, certificate: polynomial q s.t. $q^2 = p$

Certificates for Unsatisfiability

Example: $p(x) < 0$, certificate: polynomial q s.t. $q^2 = p$

Works always? Can be generalized?

Certificates for Unsatisfiability

Example: $p(x) < 0$, certificate: polynomial q s.t. $q^2 = p$

Works always? Can be generalized?

Solution to Hilbert's 17th problem:
Every polynomial that is non-negative on $\mathbb{R}^n$ is
a sum of squares of rational functions [Artin, 1927]

Certificates for Unsatisfiability

Example: $p(x) < 0$, certificate: polynomial q s.t. $q^2 = p$

Works always? Can be generalized?

Solution to Hilbert's 17th problem:
Every polynomial that is non-negative on $\mathbb{R}^n$ is
a sum of squares of rational functions [Artin, 1927]

Sums of squares of polynomials do not suffice?

Certificates for Unsatisfiability

Example: $p(x) < 0$, certificate: polynomial q s.t. $q^2 = p$

Works always? Can be generalized?

Solution to Hilbert's 17th problem:
Every polynomial that is non-negative on $\mathbb{R}^n$ is
a sum of squares of rational functions [Artin, 1927]

Sums of squares of polynomials do not suffice?

No: Motzkin form $1 + x^4y^2 + x^2y^4 - 3x^2y^2$

Certificates for Unsatisfiability

Example: $p(x) < 0$, certificate: polynomial q s.t. $q^2 = p$

Works always? Can be generalized?

Solution to Hilbert's 17th problem:
Every polynomial that is non-negative on $\mathbb{R}^n$ is
a sum of squares of rational functions [Artin, 1927]

Sums of squares of polynomials do not suffice?

No: Motzkin form $1 + x^4y^2 + x^2y^4 - 3x^2y^2$

However: all univariate polynomials, and all polynomials with degree up to 2 can be written as SOS

Special Case: System of Polynomial equations

$f_1(x) = 0, \ldots, f_r(x) = 0$ does not have a solution iff there exist

- polynomials $a_1, \ldots, a_r$, and
- sums of squares of polynomials d,

such that

$$\sum_i a_i f_i + d + 1$$

is the polynomial 0.

Special Case: System of Polynomial equations

$f_1(x) = 0, \ldots, f_r(x) = 0$ does not have a solution iff there exist

- polynomials $a_1, \ldots, a_r$, and
- sums of squares of polynomials d,

such that

$$\sum_i a_i f_i + d + 1$$

is the polynomial 0.

for a given solution, the expression cannot be zero

Special Case: System of Polynomial equations

$f_1(x) = 0, \ldots, f_r(x) = 0$ does not have a solution iff there exist

- polynomials $a_1, \ldots, a_r$, and
- sums of squares of polynomials d,

such that

$$\sum_i a_i f_i + d + 1$$

is the polynomial 0.

for a given solution, the expression cannot be zero

Special Case: System of Polynomial equations

$f_1(x) = 0, \ldots, f_r(x) = 0$ does not have a solution iff there exist

- polynomials $a_1, \ldots, a_r$, and
- sums of squares of polynomials d,

such that

$$\sum_i a_i f_i + d + 1$$

is the polynomial 0.

for a given solution, the expression cannot be zero

Example: $f_0 \equiv 1$:

Special Case: System of Polynomial equations

$f_1(x) = 0, \ldots, f_r(x) = 0$ does not have a solution iff there exist

- polynomials $a_1, \ldots, a_r$, and
- sums of squares of polynomials d,

such that

$$\sum_i a_i f_i + d + 1$$

is the polynomial 0.

for a given solution, the expression cannot be zero

Example: $f_0 \equiv 1$: $a_0 \equiv -1$, $d \equiv 0$

Special Case: System of Polynomial equations

$f_1(x) = 0, \ldots, f_r(x) = 0$ does not have a solution iff there exist

- polynomials $a_1, \ldots, a_r$, and
- sums of squares of polynomials d,

such that

$$\sum_i a_i f_i + d + 1$$

is the polynomial 0.

for a given solution, the expression cannot be zero

Example: $f_0 \equiv 1$: $a_0 \equiv -1$, $d \equiv 0$

System of polynomial equations and inequalities:
Positivstellensatz [Stengle, 1974]

Discussion

Several further interesting and widely used special cases (e.g., S-procedure)

Discussion

Several further interesting and widely used special cases (e.g., S-procedure)

How to compute such certificates?

Discussion

Several further interesting and widely used special cases (e.g., S-procedure)

How to compute such certificates?

- choose template polynomials $\sum a_i \vec{x}_i$
- solve for the coefficients (in polynomial time, using SDP) [Parrilo, 2000]

Discussion

Several further interesting and widely used special cases (e.g., S-procedure)

How to compute such certificates?

- choose template polynomials $\sum a_i \vec{x}_i$
- solve for the coefficients (in polynomial time, using SDP) [Parrilo, 2000]

Necessary degree of template polynomials:

- in linear case: 0 (Farkas Lemma), we just have to solve a linear problem
- otherwise: may be huge! usually is incrementally increased

Discussion

Several further interesting and widely used special cases (e.g., S-procedure)

How to compute such certificates?

- choose template polynomials $\sum a_i \vec{x}_i$
- solve for the coefficients (in polynomial time, using SDP) [Parrilo, 2000]

Necessary degree of template polynomials:

- in linear case: 0 (Farkas Lemma), we just have to solve a linear problem
- otherwise: may be huge! usually is incrementally increased

What about certificates after adding sin, . . . ?

Decidability and Complexity

Theorem (A. Tarski, 1930ies): $\mathit{FO}(\mathbb{R}, =, <, +, \times)$ allows quantifier elimination, and hence is decidable.

Decidability and Complexity

Theorem (A. Tarski, 1930ies): $FO(\mathbb{R}, =, <, +, \times)$ allows quantifier elimination, and hence is decidable.

However: doubly exponential in number of quantifier alternations, exponential in number of variables [Davenport and Heintz, 1988, Weispfenning, 1988]

Decidability and Complexity

Theorem (A. Tarski, 1930ies): $FO(\mathbb{R}, =, <, +, \times)$ allows quantifier elimination, and hence is decidable.

However: doubly exponential in number of quantifier alternations, exponential in number of variables [Davenport and Heintz, 1988, Weispfenning, 1988]

What about $FO(\mathbb{R}, =, <, +, \times, \sin)$?

Decidability and Complexity

Theorem (A. Tarski, 1930ies): $FO(\mathbb{R}, =, <, +, \times)$ allows quantifier elimination, and hence is decidable.

However: doubly exponential in number of quantifier alternations, exponential in number of variables [Davenport and Heintz, 1988, Weispfenning, 1988]

What about $FO(\mathbb{R}, =, <, +, \times, \sin)$?

undecidable (would allow encoding of polynomial Diophantine equations, whose solution undecidable [Matiyasevich, 1970])

Decidability and Complexity

Theorem (A. Tarski, 1930ies): $FO(\mathbb{R}, =, <, +, \times)$ allows quantifier elimination, and hence is decidable.

However: doubly exponential in number of quantifier alternations, exponential in number of variables [Davenport and Heintz, 1988, Weispfenning, 1988]

What about $FO(\mathbb{R}, =, <, +, \times, \sin)$?

undecidable (would allow encoding of polynomial Diophantine equations, whose solution undecidable [Matiyasevich, 1970])

Even equivalence of terms to zero is undecidable [Caviness, 1970], and hence also equivalence of terms (so, limited symbolic computation etc., no Nelson-Oppen, no Positivstellensatz-type certificates,)

Decidability and Complexity

Theorem (A. Tarski, 1930ies): $FO(\mathbb{R}, =, <, +, \times)$ allows quantifier elimination, and hence is decidable.

However: doubly exponential in number of quantifier alternations, exponential in number of variables [Davenport and Heintz, 1988, Weispfenning, 1988]

What about $FO(\mathbb{R}, =, <, +, \times, \sin)$?

undecidable (would allow encoding of polynomial Diophantine equations, whose solution undecidable [Matiyasevich, 1970])

Even equivalence of terms to zero is undecidable [Caviness, 1970], and hence also equivalence of terms (so, limited symbolic computation etc., no Nelson-Oppen, no Positivstellensatz-type certificates,)

Situation hopeless?

Quasi-decidability: Motivation

No algorithm that terminates for all problem instances.

Quasi-decidability: Motivation

No algorithm that terminates for all problem instances.

Algorithm that terminates for all interesting problem instances?

Quasi-decidability: Motivation

No algorithm that terminates for all problem instances.

Algorithm that terminates for all interesting problem instances?

"Interesting"?

Quasi-decidability: Motivation

No algorithm that terminates for all problem instances.

Algorithm that terminates for all interesting problem instances?

"Interesting"?

Observation: model only reflects reality up to perturbations

"interesting": satisfiability does not change under such perturbations

Quasi-decidability: Motivation

No algorithm that terminates for all problem instances.

Algorithm that terminates for all interesting problem instances?

"Interesting"?

Observation: model only reflects reality up to perturbations

“interesting”: satisfiability does not change under such perturbations

Well known in numerical analysis (well-posed problems), but in the context of decidability questions new (independently introduced by several people since ~ 2000, usually called *robust problem*).

Quasi-decidability: Definition

Constraints:

Quasi-decidability: Definition

Constraints:

$x^2 \leq 0$ $\qquad\qquad$ $x^2 \leq -0.00001$

Quasi-decidability: Definition

Constraints:

$x^2 \leq 0$ $\qquad$ $x^2 \leq -0.00001$: not robust

Quasi-decidability: Definition

Constraints:

$x^2 \leq 0$	$x^2 \leq -0.00001$: not robust
$x^2 \leq 1$	$x^2 \leq 1.00001$

Quasi-decidability: Definition

Constraints:

$x^2 \leq 0$ $\qquad$ $x^2 \leq -0.00001$: not robust

$x^2 \leq 1$ $\qquad$ $x^2 \leq 1.00001$: robust

Quasi-decidability: Definition

Constraints:

$x^2 \leq 0$ $\qquad$ $x^2 \leq -0.00001$: not robust

$x^2 \leq 1$ $\qquad$ $x^2 \leq 1.00001$: robust

$d(\phi, \phi')$: if same up to constants then maximal distance of constant, otherwise ∞

Constraint ϕ *robust* iff
there is an ε such that
for all ϕ' with $d(\phi, \phi') \leq \varepsilon$, ϕ and ϕ' are equi-satisfiable

Quasi-decidability: Definition

Constraints:

$x^2 \leq 0$	$x^2 \leq -0.00001$: not robust
$x^2 \leq 1$	$x^2 \leq 1.00001$: robust

$d(\phi, \phi')$: if same up to constants then maximal distance of constant, otherwise ∞

Constraint ϕ *robust* iff
there is an ε such that
for all ϕ' with $d(\phi, \phi') \leq \varepsilon$, ϕ and ϕ' are equi-satisfiable

Problem *quasi-decidable* iff
there is an algorithm that
correctly checks satisfiability and
terminates for all robust problem instances.

Quasi-decidability of $\mathbb{R}$

Theorem (Ratschan [2002, 2006])

$FO(\mathbb{R}, =, <, +, \times, \exp, \sin, \ldots)$ *is quasi-decidable.*

Assumptions:

- all variables bounded
- $f = 0$ shortcut for $f \leq 0 \wedge f \geq 0$

Quasi-decidability of $\mathbb{R}$

Theorem (Ratschan [2002, 2006])

$FO(\mathbb{R}, =, <, +, \times, \exp, \sin, \ldots)$ *is quasi-decidable.*

Assumptions:

- all variables bounded
- $f = 0$ shortcut for $f \leq 0 \wedge f \geq 0$

Implementation: `http://rsolver.sourceforge.net`

Methods (Quantifier-Free Case)

Special algorithms for `sat` and for `unsat`! Why?

Methods (Quantifier-Free Case)

Special algorithms for `sat` and for `unsat`! Why?

due to undecidability

- failure to prove `sat`, does not imply `unsat`, and vice versa

Methods (Quantifier-Free Case)

Special algorithms for `sat` and for `unsat`! Why?

due to undecidability

failure to prove `sat`, does not imply `unsat`, and vice versa

satisfiability: statement over one valuation,
good search method suffices (e.g., Newton's method)

approximation errors (e.g., due to rounding errors) during search
o.k., formal a-posteriori verification [Neumaier, 1990]

Methods (Quantifier-Free Case)

Special algorithms for `sat` and for `unsat`! Why?

due to undecidability

failure to prove `sat`, does not imply `unsat`, and vice versa

satisfiability: statement over one valuation,
good search method suffices (e.g., Newton's method)

approximation errors (e.g., due to rounding errors) during search
o.k., formal a-posteriori verification [Neumaier, 1990]

non-satisfiability: statement over uncountable set,
symbolic representation needed

Branch and Bound

assumption: bounded domain B for variables (e.g., $I_1 \times \cdots \times I_n$)

Branch and Bound

assumption: bounded domain B for variables (e.g., $I_1 \times \cdots \times I_n$)

$\text{test}(\phi, B) \in \{\texttt{unsat}, \texttt{unknown}\}$

Branch and Bound

assumption: bounded domain B for variables (e.g., $I_1 \times \cdots \times I_n$)

test$(\phi,\ B) \in \{\texttt{unsat}, \texttt{unknown}\}$

Algorithm $BB(\phi, B)$: either returns $\texttt{unsat}$ or runs forever

$S \leftarrow test(\phi, B)$
if $S = \texttt{unsat}$ **then** S
else
  let B be such that $B = B_1 \cup B_2$,
  non-overlapping
  if $BB(\phi, B_1) = BB(\phi, B_2) = \texttt{unsat}$ **then** $\texttt{unsat}$

Branch and Bound

assumption: bounded domain B for variables (e.g., $I_1 \times \cdots \times I_n$)

test$(\phi,\ B) \in \{\mathtt{unsat}, \mathtt{unknown}\}$

Algorithm $BB(\phi, B)$: either returns `unsat` or runs forever

$S \leftarrow test(\phi, B)$
if $S = \mathtt{unsat}$ **then** S
else
  let B be such that $B = B_1 \cup B_2$,
  non-overlapping
  if $BB(\phi, B_1) = BB(\phi, B_2) = \mathtt{unsat}$ **then** `unsat`

Can be interleaved with a satisfiability test.

unsat test

Special case: one single equality

unsat test

Special case: one single equality

Input: $f(x_1, \ldots, x_n) = 0$, intervals $I_1, \ldots, I_n$

unsat test

Special case: one single equality

Input: $f(x_1, \ldots, x_n) = 0$, intervals $I_1, \ldots, I_n$

Interval arithmetic computes interval $f(I_1, \ldots, I_n)$ such that $\{f(x_1, \ldots, x_n) \mid x_1 \in I_1, \ldots, x_n \in I_n\} \subseteq f(I_1, \ldots, I_n)$

`unsat` test

Special case: one single equality

Input: $f(x_1, \ldots, x_n) = 0$, intervals $I_1, \ldots, I_n$

Interval arithmetic computes interval $f(I_1, \ldots, I_n)$ such that $\{f(x_1, \ldots, x_n) \mid x_1 \in I_1, \ldots, x_n \in I_n\} \subseteq f(I_1, \ldots, I_n)$

if $0 \notin f(I_1, \ldots, I_n)$ **then** `unsat` **else** `unknown`

unsat test

Special case: one single equality

Input: $f(x_1, \ldots, x_n) = 0$, intervals $I_1, \ldots, I_n$

Interval arithmetic computes interval $f(I_1, \ldots, I_n)$ such that
$\{f(x_1, \ldots, x_n) \mid x_1 \in I_1, \ldots, x_n \in I_n\} \subseteq f(I_1, \ldots, I_n)$

if $0 \notin f(I_1, \ldots, I_n)$ **then** `unsat` **else** `unknown`

More powerful techniques based on

- advanced interval techniques [Neumaier, 1990, Moore et al., 2009],
- constraint propagation [Cleary, 1987, Jaulin et al., 2001],
- LP-relaxations [McCormick, 1976, Neumaier, 2004]

Challenges

In decidable polynomial case, many symbolic techniques available (Gröbner basis computation, resultants, . . .).
Sometimes efficient, combine [Passmore and Jackson, 2009].

Challenges

In decidable polynomial case, many symbolic techniques available (Gröbner basis computation, resultants, . . .).
Sometimes efficient, combine [Passmore and Jackson, 2009].

Traditionally, computer science does not take into account perturbation, and assumes decision procedures.

Use quasi-decision procedures, that is, algorithms that need not terminate for non-robust inputs.

Literature I

E. Artin. Über die Zerlegung definiter Funktionen in Quadrate. *Hamb. Abh.*, 5:100–115, 1927.

B. F. Caviness. On canonical forms and simplification. *J. ACM*, 17 (2):385–396, 1970. ISSN 0004-5411. doi: http://doi.acm.org/10.1145/321574.321591.

J. G. Cleary. Logical arithmetic. *Future Computing Systems*, 2(2): 125–149, 1987.

J. H. Davenport and J. Heintz. Real quantifier elimination is doubly exponential. *Journal of Symbolic Computation*, 5:29–35, 1988.

Luc Jaulin, Michel Kieffer, Olivier Didrit, and Éric Walter. *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer, Berlin, 2001.

Yuri Matiyasevich. Enumerable sets are diophantine. *Doklady Akademii Nauk SSSR*, 191:279–282, 1970.

Literature II

Garth P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I — convex underestimating problems. *Mathematical Programming*, 10(1): 147–175, 1976.

Ramon E. Moore, R. Baker Kearfott, and Michael J. Cloud. *Introduction to Interval Analysis*. SIAM, 2009.

Arnold Neumaier. Complete search in continuous global optimization and constraint satisfaction. *Acta Numerica*, 2004.

Arnold Neumaier. *Interval Methods for Systems of Equations*. Cambridge Univ. Press, Cambridge, 1990.

Pablo Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis, California Institute of Technology, 2000.

Grant Olney Passmore and Paul B. Jackson. Combined decision techniques for the existential theory of the reals. In *Intelligent Computer Mathematics*, 2009.

Literature III

Stefan Ratschan. Continuous first-order constraint satisfaction. In J. Calmet, B. Benhamou, O. Caprotti, L. Henocque, and V. Sorge, editors, *Artificial Intelligence, Automated Reasoning, and Symbolic Computation*, number 2385 in LNCS, pages 181–195. Springer, 2002.

Stefan Ratschan. Efficient solving of quantified inequality constraints over the real numbers. *ACM Transactions on Computational Logic*, 7(4):723–748, 2006.

M.-F. Roy and A. Szpirglas. Complexity of computation of real algebraic numbers. *Journal of Symbolic Computation*, 10:39–51, 1990.

Gilbert Stengle. A Nullstellensatz and a Positivstellensatz in semialgebraic geometry. *Mathematische Annalen*, 207(2):87–97, 1974.

Volker Weispfenning. The complexity of linear problems in fields. *Journal of Symbolic Computation*, 5(1–2):3–27, 1988.