Solving equations and inequalities using validated numeric methods

Adam Strzebonski, Wolfram Research Inc., USA
adams@wolfram.com

Abstract

Validated numeric methods can be used to significantly speed up exact solving of equations and inequalities and to find and represent exact solutions of problems that cannot be solved by known purely symbolic algorithms. Mathematica solvers use such methods to find and represent roots of univariate polynomial and transcendental functions. The cylindrical algebraic decomposition algorithm, used to solve systems of real polynomial equations and inequalities, applies validated numeric methods in the lifting phase. In my talk I will discuss the types of validation of numeric computation results used by the Mathematica solvers. I will present the validation criteria used and describe the algorithms using them.

Problem

Given an expression representing a holomorphic function $f:\mathbb{C}\supseteq U\to\mathbb{C}$ find the solutions of $f(x)==0$ in a closed rectangle $R\subset U$.

- **A symbolic-numeric heuristic**

  SolveHeuristic(*f*, *R*, *prec*)
  (1) Use a black-box numeric method with working precision *prec* to find approximate solutions.
  (2) Apply symbolic criteria to the approximate solutions in an attempt to find exact isolating sets and multiplicities of the solutions and to prove that we have found all solutions.
  (3) If (2) succeeds return a symbolic representation of the solutions otherwise return *failed*.

- **A symbolic-numeric algorithm**

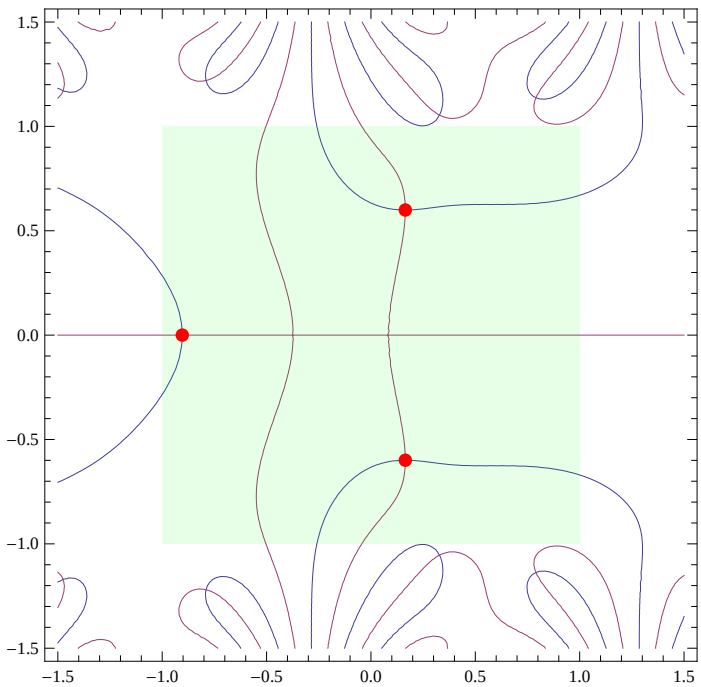
  If certain additional conditions are satisfied (shown in blue text on the following slides) the following algorithm finds the solutions.

  SolveAlgorithm(*f*, *R*)
  (1) Set *prec=machine double precision*, *answer=failed*,
  (2) While (*answer==failed*)
  (a) Set *answer*=**SolveHeuristic**(*f*, *R*, *prec*).
  (b) Set *prec*=2**prec*.
  (3) Return *answer*.

Example

Find the solutions of $3x^3 + e^x - \sin(\cos(2x-1)) + 1 = 0$ in the rectangle $[-1, 1]\times[-1, 1]$.

- **Picture**



- **Solution**

```
Solve[e^x - Sin[Cos[2 x - 1]] + 3 x^3 + 1 == 0 && -1 ≤ Re[x] ≤ 1 && -1 ≤ Im[x] ≤ 1, x]
```

```
{{x → Root[{1 + e^#1 - Sin[Cos[1 - 2 #1]] + 3 #1^3 &, -0.9038750967250084208227534835 50}]},
 {x → Root[{1 + e^#1 - Sin[Cos[1 - 2 #1]] + 3 #1^3 &,
     0.16467119796087211763838020431533578139887826330063434530726 -
      0.59938806354617993049447862473622069756099787976942345332526 i}]},
 {x → Root[{1 + e^#1 - Sin[Cos[1 - 2 #1]] + 3 #1^3 &,
     0.16467119796087211763838020431533578139887826330063434530726 +
      0.59938806354617993049447862473622069756099787976942345332526 i}]}}
```

The solutions represent exact numbers.

```
x /. %[[1]]
```

```
Root[{1 + e^#1 - Sin[Cos[1 - 2 #1]] + 3 #1^3 &, -0.9038750967250084208227534835 50}]
```

```
N[%, 100]
```

```
-0.9038750967250084208227534835495516542032068160774440826606686279418958833920882345
 1299228804012656 27
```

Assumptions

Given an expression representing a holomorphic function $f : \mathbb{C} \supseteq U \to \mathbb{C}$ find the solutions of $f(x) == 0$ in a closed rectangle $R \subset U$.

We assume that for any $n \in \mathbb{N}$

• an expression representing $f^{(n)}$ can be computed,

• an interval-arithmetic evaluation is available for $f^{(n)}$.

Interval-arithmetic evaluation

A complex interval is an element of the set $\mathbb{D} := \{(c, r) : c \in \mathbb{Q}[\mathrm{i}],\ r \in \mathbb{Q}_+ \bigcup \{\infty\}\}$.

A complex interval $I == (c(I), r(I)) \in \mathbb{D}$ represents the set $I_* == \{x \in \mathbb{C} : \ |x - c(I)| \ \leq \ r(I)\}$.

Let g be an expression representing a holomorphic function. An interval-arithmetic evaluation for g is an algorithm E_g which for any $I \in \mathbb{D}$ computes $E_g(I) \in \mathbb{D}$ with the following properties:

- $g(I_*) \subseteq E_g(I)_*$
- $I_* \subseteq J_* \Longrightarrow E_g(I)_* \subseteq E_g(J)_*$
- For any $c \in D(g)$ and $\epsilon > 0$ there exists $\delta > 0$ such that

$$r < \delta \bigwedge (d, s) == E_g((c, r)) \Longrightarrow s < \epsilon$$

Approximate solutions

Given an expression representing a holomorphic function $f : \mathbb{C} \supseteq U \to \mathbb{C}$ find the solutions of $f(x) == 0$ in a closed rectangle $R \subset U$.

Let $\xi_1, \ldots, \xi_k$ be the distinct solutions of $f(x) == 0$ in R.

We assume that there is an algorithm **ApproximateSolutions** such that

- **ApproximateSolutions**(f, R, $prec$) gives a list of numbers in R.
- For $prec \geq p_0$ card(**ApproximateSolutions**(f, R, $prec$)) $== k$
- For any $\epsilon > 0$ there exists $p_\epsilon \geq p_0$ such that for any $prec \geq p_\epsilon$ **ApproximateSolutions**(f, R, $prec$) $== \{r_1, \ldots, r_k\}$ and $\max_{1 \leq i \leq k} |r_i - \xi_i| < \epsilon$.

Root existence criterion

For $I \in \mathbb{D}$ let us denote $I_{\min} := \min_{z \in I_*} |z|$ and $I_{\max} := \max_{z \in I_*} |z|$.

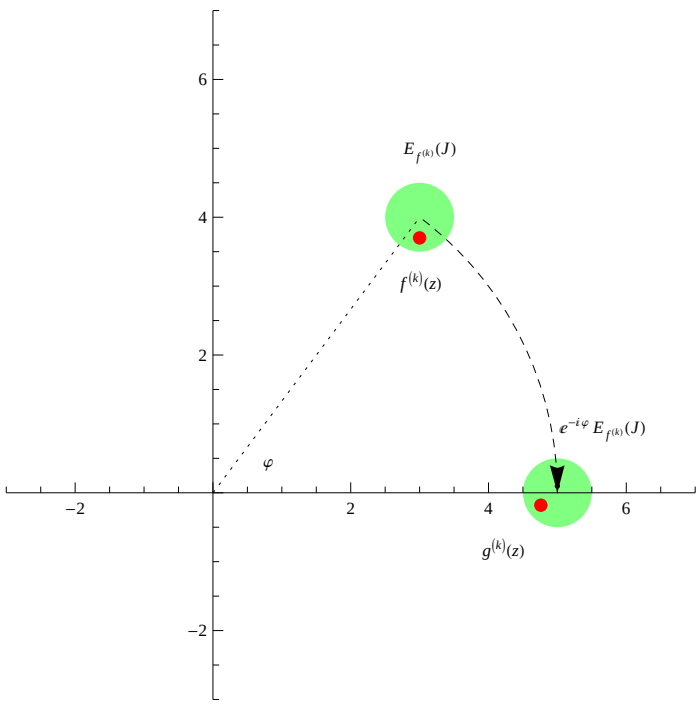
Criterion: Let $f : \mathbb{C} \supseteq U \to \mathbb{C}$ be a holomorphic function, let $I \in \mathbb{D}$ be such that $I_* \subset U$. Suppose that, for some $k \in \mathbb{N}_+$ and $J == (c(I), r) \in \mathbb{D}$, $J_* \subset U$

$$\frac{\left(E_{f^{(k)}}(J)\right)_{\min}}{k!} r^k > \sum_{i=0}^{k-1} \frac{\left(E_{f^{(i)}}(I)\right)_{\max}}{i!} r^i$$

then f has exactly k roots in J_* (counted with multiplicities).

- **Proof**

Let $z_0 := c(I)$, let $\varphi := \operatorname{Arg}\left(c\left(E_{f^{(k)}}(J)\right)\right)$ and let $g(z) := \exp(-i\,\varphi)\, f(z)$.



For $z \in J_*$ we have

$$\left|\operatorname{Re}\left(\frac{g^{(k)}(z)}{k!}\right)\right| \geq \frac{\left(E_{f^{(k)}}(J)\right)_{\min}}{k!} > \sum_{i=0}^{k-1} \frac{\left(E_{f^{(i)}}(I)\right)_{\max}}{i!} r^{i-k} \geq \sum_{i=0}^{k-1} \left|\frac{f^{(i)}(z_0)}{i!}\right| r^{i-k} = \sum_{i=0}^{k-1} \left|\frac{g^{(i)}(z_0)}{i!}\right| r^{i-k}$$

The criterion follows from applying the following theorem to g and J_*.

Theorem (Neumaier, 1988) Let $f : \mathbb{C} \supseteq U \to \mathbb{C}$ be a holomorphic function and let $z_0 \in U$ and $r > 0$ be such that $D := \{z : |z - z_0| \leq r\} \subset U$. If for all $z \in D$

$$\left|\operatorname{Re}\left(\frac{f^{(k)}(z)}{k!}\right)\right| > \sum_{i=0}^{k-1} \left|\frac{f^{(i)}(z_0)}{i!}\right| r^{i-k}$$

then f has exactly k roots in D (counted with multiplicities).

Application of the root existence criterion

Criterion: Let $f : \mathbb{C} \supseteq U \to \mathbb{C}$ be a holomorphic function, let $I \in \mathbb{D}$ be such that $I_* \subset U$. Suppose that, for some $k \in \mathbb{N}_+$ and $J == (c(I), r) \in \mathbb{D}$, $J_* \subset U$

$$\frac{\left(E_{f^{(k)}}(J)\right)_{\min}}{k!} r^k > \sum_{i=0}^{k-1} \frac{\left(E_{f^{(i)}}(I)\right)_{\max}}{i!} r^i$$

then f has exactly k roots in J_* (counted with multiplicities).

We attempt to find an isolated solution of $f == 0$ starting with an approximate solution $z_0 \in$ **ApproximateSolutions**$(f, R, prec)$. If successful, we get $r > 0$ and $k \in \mathbb{N}_+$ such that $D(z_0, r)$ contains exactly k solutions of $f == 0$. If $k > 1$ the method cannot distinguish between one solution of multiplicity k and a cluster of solutions. If f is an elementary function one could use zero-testing to prove multiplicity of the solution (the currently known zero-testing algorithm [Richardson] relies on Schanuel's conjecture for proof of termination).

(1) Set $k := 1$. If $z_0 \neq 0$ set $\epsilon := \left| z_0 \right| 10^{-2\,\mathrm{prec}}$ else set $\epsilon := 10^{-2\,\mathrm{prec}}$. Put $I := (z_0, \epsilon)$ and $M_0 := \frac{\left(E_f(I)\right)_{\max}}{i!}$.

(2) While $k \leq k_{\max}$ do

(a) Compute $m_k := \frac{\left(E_{f^{(k)}}(I)\right)_{\min}}{k!}$ and $M_k := \frac{\left(E_{f^{(k)}}(I)\right)_{\max}}{k!}$.

(b) If $m_k == 0$, increment k and continue the loop.

(c) Set $r := \max_{0 \leq i \leq k-1} \left(2\, k\, \frac{M_i}{m_k}\right)^{\frac{1}{k-i}}$.

(d) If $\frac{\left(E_{f^{(k)}}((z_0, r))\right)_{\min}}{k!} r^k > \sum_{i=0}^{k-1} M_i\, r^i$ return (r, k).

(e) Increment k and continue the loop.

(3) Return *failed*.

$k_{\max}$ is a bound on the total number of solutions, if known, otherwise it is a fixed parameter. ϵ and r are rational numbers -- floating point number approximations of the values given in (1) and (2c). If z_0 is close to a root ξ of f of multiplicity μ, $k \leq \mu$ and $\frac{m_k}{M_k} \approx 1$ then $r \lesssim 2\, \mu^2 \left| z_0 - \xi \right|$. Hence if for some k, $\frac{m_k}{M_k} \approx 1$ and r is large we can return *failed* without continuing the loop all the way to $k_{\max}$.

Verifying the number of solutions

Given an expression representing a holomorphic function $f : \mathbb{C} \supseteq U \to \mathbb{C}$ find the solutions of $f(x) == 0$ in a closed rectangle $R \subset U$.

To find the number of solutions of $f(x) == 0$ in R (counted with multiplicities) we compute the winding number of f along the boundary of R. We subdivide the boundary into segments such that for each segment either $0 \notin \mathrm{Re}(E_f(I))$ or $0 \notin \mathrm{Im}(E_f(I))$. This requires that there are no zeros of f on the boundary of R. We use rectangular interval arithmetic here.

Finding approximate solutions

ApproximateSolutions(*f*, *R*, *prec*) uses either computational geometry methods or the Kravanja-Van Barel algorithm to find machine double precision solutions and uses them as starting points to the Newton method. Satisfying the **SolveAlgorithm** conditions would require an arbitrary-precision version of the part finding the starting points (currently not implemented in *Mathematica*).

Semialgebraic sets

A real polynomial condition is a formula

$f(x_1, \ldots, x_n)\, \rho\, 0$

where $f \in \mathbb{R}[x_1, \ldots, x_n]$ and ρ is one of $<$, $\leq$, $>$, $\geq$, $=$, or $\neq$.

A quantified real polynomial formula is a formula constructed with real polynomial conditions using Boolean operators and quantifiers over real variables.

A subset of $\mathbb{R}^n$ is semialgebraic if it is a solution set of a real polynomial formula with n free variables.

Cell decomposition

Every semialgebraic set can be represented as a finite union of disjoint cells defined recursively as follows.

• A cell in $\mathbb{R}$ is a point or an open interval.

• A cell in $\mathbb{R}^k$ has one of the two forms

$\{(a_1, \ldots, a_k, a_{k+1}) : (a_1, \ldots, a_k) \in C_k \bigwedge a_{k+1} == r(a_1, \ldots, a_k)\}$

$\{(a_1, \ldots, a_k, a_{k+1}) : (a_1, \ldots, a_k) \in C_k \bigwedge r_1(a_1, \ldots, a_k) < a_{k+1} < r_2(a_1, \ldots, a_k)\}$

where C_k is a cell in $\mathbb{R}^k$, r is a continuous algebraic function, and r_1 and r_2 are continuous algebraic functions, $-\infty$, or ∞, and $r_1 < r_2$ on C_k.

By an algebraic function we mean a function

$\text{Root}_{x_{k+1},p}\ f : C_k \ni (x_1, \ ..., \ x_k) \longrightarrow \text{Root}_{x_{k+1},p}\ f(x_1, \ ..., \ x_k) \in \mathbb{R}$

where

$f = c_0\, x_{k+1}^m + c_1\, x_{k+1}^{m-1} + \ldots + c_m \in \mathbb{R}[x_1, \ldots, x_k, x_{k+1}]$

is a polynomial and

$\text{Root}_{x_{k+1},p}\ f(x_1, \ ..., \ x_k)$

is the p-th real root (multiplicities counted) of f treated as a univariate polynomial in x_{k+1}.

Cylindrical algebraic decomposition

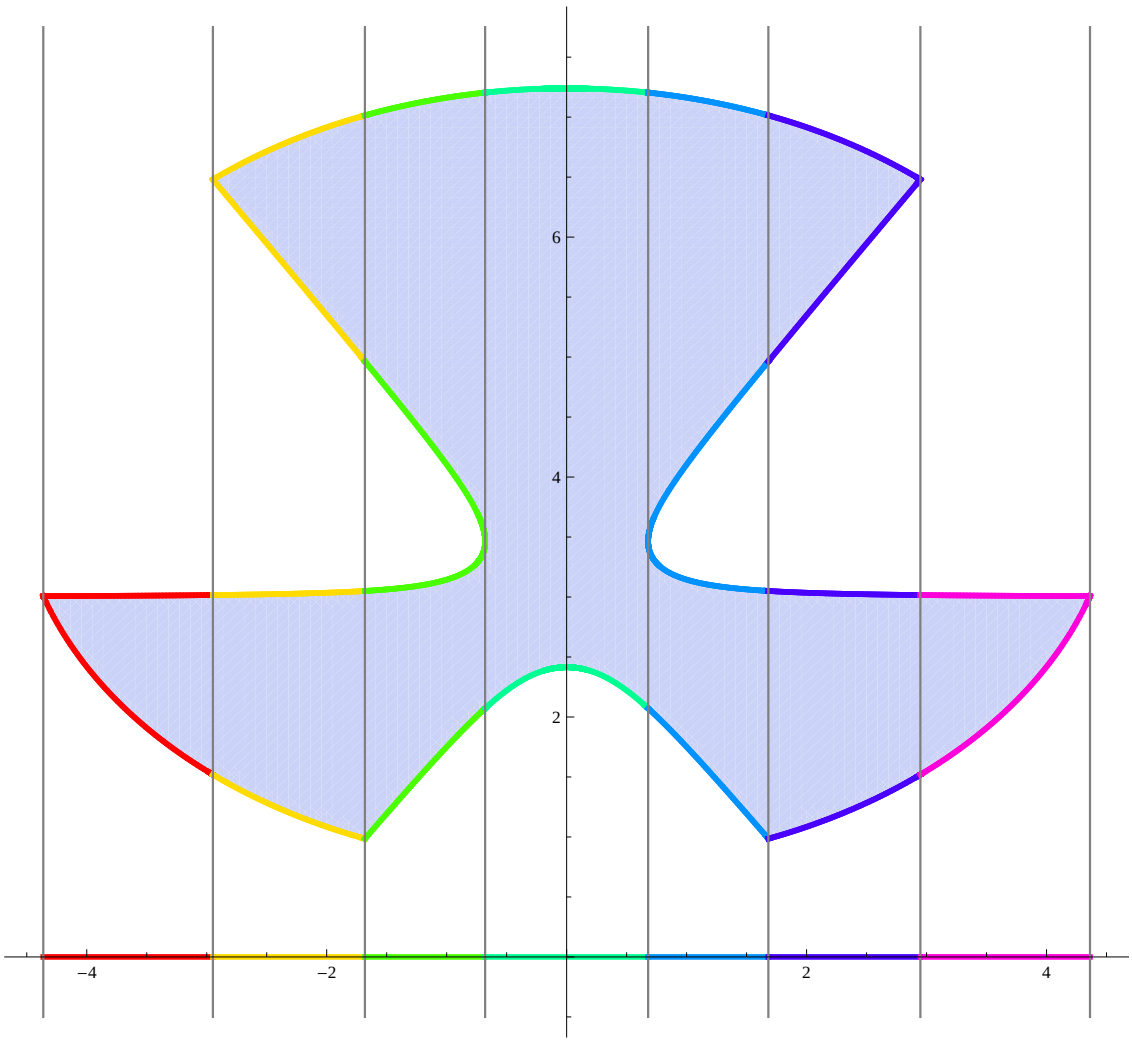
The CAD algorithm, introduced by Collins (1975), allows to compute a cell decomposition of any semialgebraic set presented by a real polynomial formula.

• Cells are arranged cylindrically.

• Polynomials whose roots bound cells extending C are delineable over C.

A set $A \subset \mathbb{R}[x_1, \ldots, x_k, x_{k+1}]$ is delineable over $C \subseteq \mathbb{R}^k$ if each $f \in A$ has a fixed number of real roots on C as a polynomial in x_{k+1}, the roots are continuous functions on C, they have constant multiplicities, and two roots of $f, g \in A$ are equal either everywhere or nowhere in C.

```
ineqs = -5 (y - 3)^3 + 7 x^2 (y - 3) < 1 && x^2 + 2 (y - 4)^2 < 21;
```



The projection phase of the CAD algorithm

Finding a cell decomposition of a semialgebraic set using the CAD algorithm consists of two phases, projection and lifting. In the projection phase we start with the set A_n of factors of the polynomials present in the system, and eliminate variables one by one using a projection operator P such that

$$P_{k+1} : \mathbb{R}[x_1, \ldots, x_k, x_{k+1}] \supset A_{k+1} \longrightarrow A_k \subset \mathbb{R}[x_1, \ldots, x_k]$$

and, generally speaking, if all polynomials of A_k have constant signs on a cell $C \subset \mathbb{R}^k$, then all polynomials of A_{k+1} are delineable over C. This way the roots of polynomials of $A_1, \ldots, A_n$ are the algebraic functions needed in the construction of the cell decomposition of the semialgebraic set.

The lifting phase of the CAD algorithm

In the lifting phase we find a cell decomposition of the semialgebraic set. We start with cells in $\mathbb{R}^1$ consisting of all distinct roots of A_1 and the open intervals between the roots. We find a sample point in each of the cells and remove the cells whose sample points do not satisfy the system describing the semialgebraic set (the system may contain conditions involving only x_1). Now we lift the cells to cells in $\mathbb{R}^n$, one dimension at a time. Suppose we have lifted the cells to $\mathbb{R}^k$. To lift a cell $C \subset \mathbb{R}^k$ to $\mathbb{R}^{k+1}$ we find the real roots of $A_{k+1}(c)$, where c is a sample point in C and the elements of $A_{k+1}(c)$ are the elements of A_{k+1} with $x_1, \ldots x_k$ replaced with the coordinates of c. If r is the p-th root of $f(c, x_{k+1})$ for some $f \in A_{k+1}$, then $\mathrm{Root}_{x_{k+1},p}\, f(x_1, \ldots, x_k)$ is a continuous algebraic function on C, because the polynomials of A_{k+1} are delineable on C. The lifting of C to $\mathbb{R}^{k+1}$ will consist of graphs of such algebraic functions, and of the slices of $C \times \mathbb{R}$ between the consecutive graphs. The sample points in each of the new cells will be obtained by adding the $k+1$-st coordinate to c, equal to one of the roots or to a number between two consecutive roots. Similarly as in the first step we remove those lifted cells whose sample points do not satisfy the system describing the semialgebraic set.

To lift a cell we need to find the real root structure of $A_{k+1}(c)$. That is, we need to find disjoint intervals isolating the distinct real roots of $A_{k+1}(c)$, find the multiplicity of each root and identify the common roots of different elements of $A_{k+1}(c)$. The coefficients of sample points are in general algebraic numbers. We can avoid algebraic number computations by using a validated numeric method.

Problem (Real root structure)

Given

• a finite set of polynomials $A \subset \mathbb{Z}[x_1, \ldots, x_k, x_{k+1}]$, where $k \geq 0$,

• a tuple $(I_1, \ldots, I_k)$ of real intervals such that $a_i \in I_i$ for $1 \leq i \leq k$,

find

• a tuple $(J_1, \ldots, J_l)$ of disjoint real intervals,

• a function $mult : A \times \{1, \ldots, l\} \longrightarrow \mathbb{N}$

such that

• the polynomials $\{f(a, x_{k+1}) : f \in A\}$ have l distinct real roots $r_1, \ldots, r_l$,

• $r_i \in J_i$ for $1 \leq i \leq l$,

• $mult(f, i)$ is the multiplicity of r_i as a root of f for all $f \in A$ and $1 \leq i \leq l$.

■ **Additional assumptions**

For any $f, g \in A$ we can obtain the following information

• the degree of the polynomial $f(a, x_{k+1})$,

• the degree of the g.c.d. of $f(a, x_{k+1})$ and $g(a, x_{k+1})$,

• the degree of the g.c.d. of $f(a, x_{k+1})$ and $\partial_{x_{k+1}} f(a, x_{k+1})$.

■ **Obtaining the additional information in the CAD algorithm**

This information may be deduced from knowing the number of initial coefficients of $f(a, x_{k+1})$ that are zero and the number of initial principal subresultant coefficients (PSC) of $f(a, x_{k+1})$ and $g(a, x_{k+1})$ or of $f(a, x_{k+1})$ and $\partial_{x_{k+1}} f(a, x_{k+1})$ that are zero.

In the CAD algorithm, the projection A_k of the polynomial set A_{k+1} contains the nonconstant initial coefficients, the discriminants and the pairwise resultants for all polynomials in A_{k+1}. The projection may also contain PSC for pairs of polynomials or for pairs polynomial and its derivative.

The signs of projection polynomials A_k on a given cell cell $C \subset \mathbb{R}^k$ are constant. If $a \in C$ then for a polynomial $g \in A_k$, $g(a) == 0$ iff C is the graph of a root of g. Hence if a coefficient or a PSC belongs to the projection, the information on whether it is zero at a comes for free from the construction of C. We just need to keep track of where the projection polynomials come from and which projection polynomials are zero on the given cell.

If we need to know whether a PSC that is not a part of the projection is zero at a, we can use zero testing to obtain this information. This may require nontrivial computation, but still it tends to be faster than lifting the cell using exact algebraic number computation.

Timings

TABLE 1. Examples from applications (timings)

Example	Time *Proj*	Time *ACAD*	Time *ECAD*	Time *ICAD*	Time *QEPCAD*
1	0.08	0.12	3.96	3.93	0.83
$2-1$	0.22	1.94	> 3600	> 3600	389
$2-2$	0.27	2.61	> 3600	> 3600	1235
3	0.37	19.6	> 3600	> 3600	$F(3446)$
4	6.43	408	> 3600	> 3600	$F(402)$

TABLE 3. Randomly generated examples (timings)

Example	Time *Proj*	Time *ACAD*	Time *ECAD*	Time *ICAD*	Time *QEPCAD*
$a-1$	0.25	1.49	> 3600	> 3600	$F(306)$
$a-2$	0.28	2.19	> 3600	> 3600	$F(270)$
$a-3$	0.03	1.07	2123	1624	3.68
$a-4$	0.04	1.23	> 3600	> 3600	0.34
$a-5$	0.88	3.09	> 3600	> 3600	$F(211)$
$b-1$	0.05	15.7	> 3600	> 3600	$F(1272)$
$b-2$	13.4	15.5	> 3600	> 3600	$F(265)$
$b-3$	0.32	19.1	> 3600	> 3600	$F(498)$
$b-4$	14.2	24.3	> 3600	> 3600	$F(317)$
$b-5$	0.11	34.5	> 3600	> 3600	$F(3278)$
$c-1$	0.64	173	> 3600	> 3600	$F(1042)$
$c-2$	0.23	104	> 3600	> 3600	> 10000
$c-3$	1.05	196	> 3600	> 3600	$F(279)$
$c-4$	4.32	854	> 3600	> 3600	$F(1766)$
$c-5$	65.7	606	> 3600	> 3600	$F(624)$

Symbolic-numeric real root structure computation

RealRootStructureHeuristic(A, I, *prec*)
(1) Let $F := \{g(c(I_1), \ldots, c(I_k), z) : g \in A\} \subset \mathbb{Q}[z]$
(2) Use a black-box numeric method with working precision *prec* to find approximations of all complex roots of elements of F.
(3) Apply symbolic criteria to the approximate solutions in an attempt to find the real root structure of A at a.

- **A symbolic-numeric algorithm for cell lifting in CAD**

CellLifting(*projection data*)
(1) Construct cells $C_1, \ldots, C_m$ in $\mathbb{R}$ (exact real root isolation).
(2) Set *prec=initial precision*, *answer=empty*.
(3) For $1 \le i \le m$,
(a) Let a be a sample point in C_i. If $a \in \mathbb{Q}$, set $I := (a, 0)$, else find $I == (c, r)$ such that $c - r < a < c + r$ and $r < |c| 10^{-\text{prec}}$.
(b) Append **CellLiftingRecursive**(C_i, I, *prec*, *projection data*) to *answer*.
(4) Return *answer*.

CellLiftingRecursive(C, I, *prec*, *projection data*)
(1) Set $I_C := I$, *newprec=prec*.
(2) Compute *root structure* = **RealRootStructureHeuristic**(A_{k+1}, I_C, *newprec*).
(3) While (*root structure==failed*)
(a) Set *newprec* = 2 *newprec*.
(b) Recompute I_C working with precision *newprec*.
(c) Compute *root structure* = **RealRootStructureHeuristic**(A_{k+1}, I_C, *newprec*).
(4) Construct cells $C_1, \ldots, C_m$ in $\mathbb{R}^{k+1}$ and intervals $I_1, \ldots, I_m$ in $\mathbb{I}^{k+1}$ extending C and I_c.
(5) Set *answer=empty*. For $1 \le i \le m$, append **CellLiftingRecursive**(C_i, I_i, *newprec*, *projection data*) to *answer*.
(6) Return *answer*.

An alternative version of the algorithm, currently used by the *Mathematica* CAD implementation, is to revert to exact algebraic number computations for the cell C if the call to **RealRootStructureHeuristic** in step (2) of **CellLiftingRecursive** fails. The termination of the version of the algorithm presented here depends on the following property of the numeric root finding method **NumericRoots**(f, *prec*).

For any polynomial $f \in \mathbb{Q}[z]$ and any $\epsilon > 0$ there exists p_ϵ such that for any $prec \ge p_\epsilon$
NumericRoots(f, *prec*) $== \{r_1, \ldots, r_n\}$ and $\max_{1 \le i \le n} |r_i - \xi_i| < \epsilon$
where $f(z) == a_n(z - \xi_1) \ldots (z - \xi_n)$

Interval-arithmetic evaluation of polynomials

Let us slightly modify the definition of complex interval.

A complex interval is an element of the set $\mathbb{D} := \{(c, r) : c \in \mathbb{Q}[\mathbb{i}],\ r \in \mathbb{Q}_+ \bigcup \{0, \infty\}\}$.

A real interval is an element of the set $\mathbb{I} := \{(c, r) : c \in \mathbb{Q},\ r \in \mathbb{Q}_+ \bigcup \{0, \infty\}\}$. We have $\mathbb{I} \subset \mathbb{D}$.

We allow one-point intervals here -- unlike for analytic functions, we can always compute the exact value of a rational polynomial at a Gaussian rational point.

For $I == (I_1, \ldots, I_k) \in \mathbb{D}^k$ let $I_* := (I_1)_* \times \ldots \times (I_k)_* \subseteq \mathbb{C}^k$.

Let $g \in \mathbb{Q}[x_1, \ldots, x_k]$. An interval-arithmetic evaluation for g is an algorithm E_g which for any $I \in \mathbb{D}^k$ computes $E_g(I) \in \mathbb{D}$ with the following properties:

• $g_*(I_*) \subseteq E_g(I)_*$

• $I_* \subseteq J_* \Longrightarrow E_g(I)_* \subseteq E_g(J)_*$

• For any $a \in \mathbb{C}^k$ and $\epsilon > 0$ there exists $\delta > 0$ such that

$$I == ((a_1, r_1), \ldots, (a_k, r_k)) \bigwedge \max_{1 \le i \le k} r_i < \delta \bigwedge (c, r) == E_g(I) \Longrightarrow r < \epsilon$$

• If $I \in \mathbb{I}^k$ then $E_g(I) \in \mathbb{I}$.

For $f == I_n z^n + \ldots + I_0 \in \mathbb{D}[z]$ and $z_0 \in \mathbb{Q}[\mathbb{i}]$ define

$f(z_0) := E_g(I_0, \ldots, I_n, (z_0, 0)) \in \mathbb{D}$

where $g == x_n z^n + \ldots + x_0 \in \mathbb{Q}[x_0, \ldots, x_n, z]$.

Root existence criterion

For $I \in \mathbb{D}$ let us denote $I_{\min} := \min_{z \in I_*} |z|$ and $I_{\max} := \max_{z \in I_*} |z|$.

Criterion: Let $f == I_n z^n + \ldots + I_0 \in \mathbb{D}[z]$ with $0 \notin I_n$, let $z_0 \in \mathbb{Q}[\mathbb{i}]$ and let $c_i := \frac{f^{(i)}(z_0)}{i!} \in \mathbb{D}$ for $0 \le i \le n$. Suppose that

$$\max_{0 \le i < m} \left(\frac{n\,(c_i)_{\max}}{(c_m)_{\min}} \right)^{\frac{1}{m-i}} < r < \min_{m < i \le n} \left(\frac{(c_m)_{\min}}{n (c_i)_{\max}} \right)^{\frac{1}{i-m}}$$

then for any $a_0 \in I_0, \ldots, a_n \in I_n$ the polynomial $a_n z^n + \ldots + a_0 \in \mathbb{C}[z]$ has exactly m roots in the disk $D(z_0, r)$ (counted with multiplicities).

- **Proof**

The criterion follows from:

Proposition (A.S., JSC 2006, proof based on Rouche's theorem)

Let $f \in \mathbb{C}[x]$ be a polynomial of degree n, let $z_0 \in \mathbb{C}$ and let $c_i := \left| \frac{f^{(i)}(z_0)}{i!} \right|$. Suppose that

$$\max_{0 \le i < k} \left(\frac{n\,c_i}{c_k} \right)^{\frac{1}{k-i}} < r < \min_{k < i \le n} \left(\frac{c_k}{n\,c_i} \right)^{\frac{1}{i-k}}$$

then f has exactly k roots in the disk $D(z_0, r)$ (counted with multiplicities).

Application of the root existence criterion

Criterion: Let $f == I_n z^n + \ldots + I_0 \in \mathbb{D}[z]$ with $0 \notin I_n$, let $z_0 \in \mathbb{Q}[\mathbb{i}]$ and let $c_i := \frac{f^{(i)}(z_0)}{i!} \in \mathbb{D}$ for $0 \le i \le n$. Suppose that

$$\max_{0 \le i < m}\left(\frac{n\,(c_i)_{\max}}{(c_m)_{\min}}\right)^{\frac{1}{m-i}} < r < \min_{m < i \le n}\left(\frac{(c_m)_{\min}}{n(c_i)_{\max}}\right)^{\frac{1}{i-m}}$$

then for any $a_0 \in I_0, \ldots, a_n \in I_n$ the polynomial $a_n z^n + \ldots + a_0 \in \mathbb{C}[z]$ has exactly m roots in the disk $D(z_0, r)$ (counted with multiplicities).

Problem

Given

• a finite set of polynomials $A \subset \mathbb{Z}[x_1, \ldots, x_k, x_{k+1}]$, where $k \ge 0$,

• a tuple $(I_1, \ldots, I_k)$ of real intervals such that $a_i \in I_i$ for $1 \le i \le k$,

find

• a tuple $(J_1, \ldots, J_l)$ of disjoint real intervals,

• a function $mult : A \times \{1, \ldots, l\} \longrightarrow \mathbb{N}$

such that

• the polynomials $\{f(a, x_{k+1}) : f \in A\}$ have l distinct real roots $r_1, \ldots, r_l$,

• $r_i \in J_i$ for $1 \le i \le l$,

• $mult(f, i)$ is the multiplicity of r_i as a root of f for all $f \in A$ and $1 \le i \le l$.

Find the root structure of each polynomial

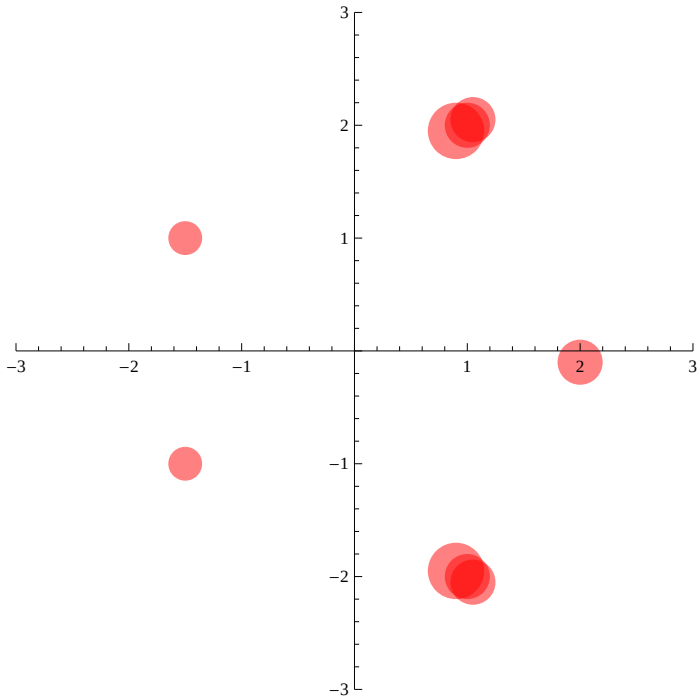
For $f \in A$ do

Find root approximations

(1) Compute $f_I == I_n z^n + \ldots + I_0 == f(I_1, \ldots, I_k, z) \in \mathbb{D}[z]$.
(2) While $n \ge 0$ and $0 \in I_n$
  (a) If the coefficient at x_{k+1}^n of $f(a, x_{k+1})$ is nonzero, return *failed*.
  (b) Set $f_I = I_{n-1} z^{n-1} + \ldots + I_0$ and decrement n.
(3) If $f_I == 0$ set $R_f^{\text{real}} = R_f^{\text{complex}} := \varnothing$ and continue the loop.
(4) Let f_c be $f(c(I_1), \ldots, c(I_k), z)$ with monomials of degree higher than n removed.
(5) Compute approximate roots $z_1, \ldots, z_n \in \mathbb{Q}[\mathbb{i}]$ of f_c.

Use the criterion to find disjoint disks containing roots of $f(a, x_{k+1})$

(6) For $1 \le i \le n$ find the smallest positive integer m_i and a radius r_i such that the criterion is satisfied for f_I, z_i, m_i and r_i.
(7) Let $R := \{(z_i, r_i, m_i) : 1 \le i \le n\}$.

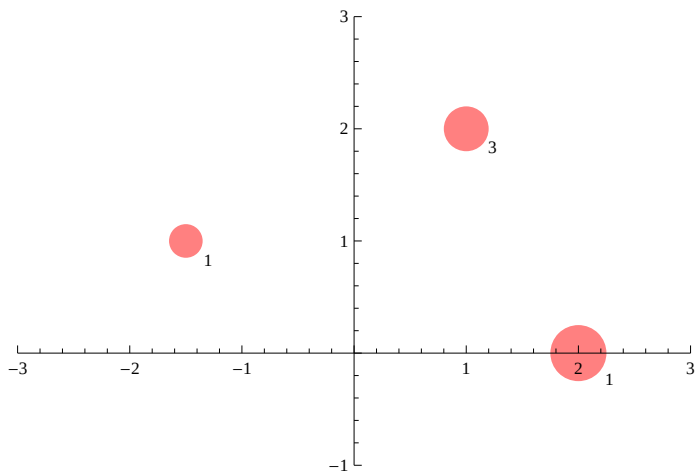


(8) For each $(z, r, m) \in R$ such that $z \notin \mathbb{R} \bigwedge D(z, r) \bigcap \mathbb{R} \neq \varnothing$

(a) Find the smallest positive integer m' and a radius r' such that the criterion is satisfied for f_I, z', m' and $\mathrm{re}(r)$.

(b) Replace (z, r, m) with $(\mathrm{re}(z), r', m')$ in R.

(8) While there exist $(z, r, m), (z', r', m') \in R$, $D(z, r) \bigcap D(z', r') \neq \varnothing$ and $r \leq r'$, remove (z', r', m') from R.



(9) Set $R_f^{\mathrm{real}} := \{(z, r, m) \in R : z \in \mathbb{R}\}$ and $R_f^{\mathrm{complex}} := \{(z, r, m) \in R : \mathrm{im}(z) > 0\}$.

- **Verify that the disks contain all roots, each element of R_f^{real} contains one root and the root is real**

(10) If $\sum_{(z,r,m)\in R_f^{\mathrm{real}}} m + 2 \sum_{(z,r,m)\in R_f^{\mathrm{complex}}} m \neq n$ return *failed*.

(11) If there exists $(z, r, m) \in R_f^{\mathrm{real}}$ such that $m > 1$ find the degree d of the g.c.d. of $f(a, x_{k+1})$ and $\partial_{x_{k+1}} f(a, x_{k+1})$.

If $\mathrm{card}\left(R_f^{\mathrm{real}}\right) + 2\,\mathrm{card}\left(R_f^{\mathrm{complex}}\right) \neq n - d$ return *failed*.

Identify the common real roots

For $f, g \in A$ such that there exist $(z, r, m) \in R_f^{\text{real}}$ and $(z', r', m') \in R_g^{\text{real}}$ with $D(z, r) \bigcap D(z', r') \neq \varnothing$ do

(1) Set $m_{\text{tot}} = 0$.
(2) For each $(z, r, m) \in R_f^{\text{real}}$ and $(z', r', m') \in R_g^{\text{real}}$ such that $D(z, r) \bigcap D(z', r') \neq \varnothing$ set $m_{\text{tot}} = m_{\text{tot}} + \min(m, m')$.
(3) For each $(z, r, m) \in R_f^{\text{complex}}$ and $(z', r', m') \in R_g^{\text{complex}}$ such that $D(z, r) \bigcap D(z', r') \neq \varnothing$ set $m_{\text{tot}} = m_{\text{tot}} + 2 \min(m, m')$.
(4) Find the degree d of the g.c.d. of $f(a, x_{k+1})$ and $g(a, x_{k+1})$.
(5) If $m_{\text{tot}} \neq d$ return *failed*.

Return isolating intervals for the real roots and the multiplicity function

(1) Find real intervals $(J_1, \ldots, J_l)$ by picking one representative from each set of intersecting intervals in $\left\{(z, r) : f \in A \bigwedge (z, r, m) \in R_f^{\text{real}}\right\}$.
(2) For $f \in A$ and $1 \leq i \leq l$, if there exists $(z, r, m) \in R_f^{\text{real}}$ such that $J_i \bigcap D(z, r) \neq \varnothing$ then $mult(f, i) = m$ else $mult(f, i) = 0$.
(3) Return $(J_1, \ldots, J_l)$ and *mult*.