An Incremental Algorithm for Computing Cylindrical Algebraic Decomposition and Its Application to Quantifier Elimination

Changbo Chen
Joint work with Marc Moreno Maza

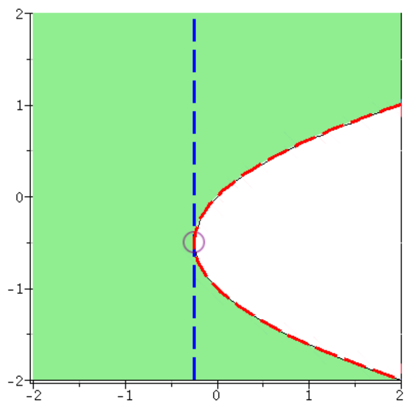
ORCCA, University of Western Ontario, Canada

Aug. 2, 2013
SIAM AG 2013, Fort Collins, Colorado, USA

Cylindrical Algebraic Decomposition (CAD) of $\mathbb{R}^n$

A CAD of $\mathbb{R}^n$ is a partition of $\mathbb{R}^n$ such that each cell in the partition is a connected semi-algebraic subset of $\mathbb{R}^n$ and all the cells are cylindrically arranged.

Two subsets A and B of $\mathbb{R}^n$ are called cylindrically arranged if for any $1 \leq k < n$, the projections of A and B on $\mathbb{R}^k$ are either equal or disjoint.



Cylindrical algebraic decomposition (CAD)

Invented by G.E. Collins in 1973 for solving Real Quantifier Elimination (QE) problems.

Previous work on CAD

Adjacency and clustering techniques (D. Arnon, G.E. Collins and S. McCallum 84), Improved projection operator (H. Hong 90; S. McCallum 88, 98; C. Brown 01), Partially built CADs (Collins and Hong 91, A. Strzeboński 00), Improved stack construction (G.E. Collins, J.R. Johnson, and W. Krandick), Efficient projection orders (A. Dolzmann, A. Seidl and T. Sturm 04), Making use of equational constraints (G.E. Collins 98; C. Brown and S. McCallum 05, R. Bradford, J. Davenport, M. England, S. McCallum and D. Wilson 12), Set-theoretical operations by CAD (A. Strzeboński 10), Computing CAD via triangular decompositions (C. Chen, M. Moreno Maza, B. Xia and L. Yang 09), ...

Software

QEPCAD, Mathematica, Redlog, SyNRAC, RegularChains (TCAD).

Outline

1. First Idea: Introduce Case Discussion
2. Second Idea: Compute the Decomposition Incrementally
3. Third Idea: Compute CAD of a Variety
4. QE via TCAD
5. Implementation and Benchmark
6. Application

Outline

CAD based on projection-lifting scheme (PCAD)

Projection

- Let $Proj$ be a projection operator.
- Repeatedly apply $Proj$:

$$F_n(x_1, \ldots, x_n) \xrightarrow{Proj} F_{n-1}(x_1, \ldots, x_{n-1}) \xrightarrow{Proj} \cdots \xrightarrow{Proj} F_1(x_1).$$

Lifting

- The real roots of the polynomials in F_1 plus the open intervals between them form an F_1-invariant CAD of $\mathbb{R}^1$.
- For each cell C of the F_{k-1} invariant CAD of $\mathbb{R}^{k-1}$, isolating the real roots of the polynomials of F_k at a sample point of C, produces all the cells of the F_k-invariant CAD of $\mathbb{R}^k$ above C.

CAD based on triangular decompositions (TCAD)

Motivation: potential drawback of Collins' scheme

- The projection operator is a function defined independently of the input system.
- As a result, a strong projection operator (Collins-Hong operator) usually produces much more polynomials than needed.
- A weak projection operator (McCalumn-Brown operator) may fail for non-generic cases.

Solution: make case discussion during projection

- Case discussion is common for algorithms computing triangular decomposition.
- At ISSAC'09, we (with B. Xia and L. Yang) introduced case discussion (as in triangular decomposition of polynomials systems) into CAD computation. As a result, the projection phase in classical CAD algorithm is replaced by computing a complex cylindrical tree.

Complex cylindrical tree

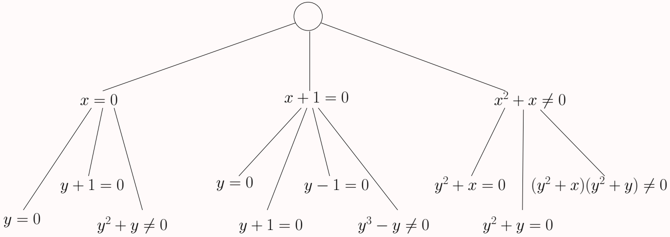
- let $\alpha = (\alpha_1, \ldots, \alpha_{n-1}) \in \mathbb{C}^{n-1}$
- define $\mathbb{C}[x_1, \ldots, x_n] \xrightarrow{\Phi_\alpha} \mathbb{C}[x_n]$, where $p(x_1, \ldots, x_n) \mapsto p(\alpha, x_n)$

Separation

Let $S \subset \mathbb{C}^{n-1}$ and $P \subset \mathbf{k}[x_1, \ldots, x_{n-1}, x_n]$ be a finite set of level n polynomials. We say that P separates above S if for each $\alpha \in S$:

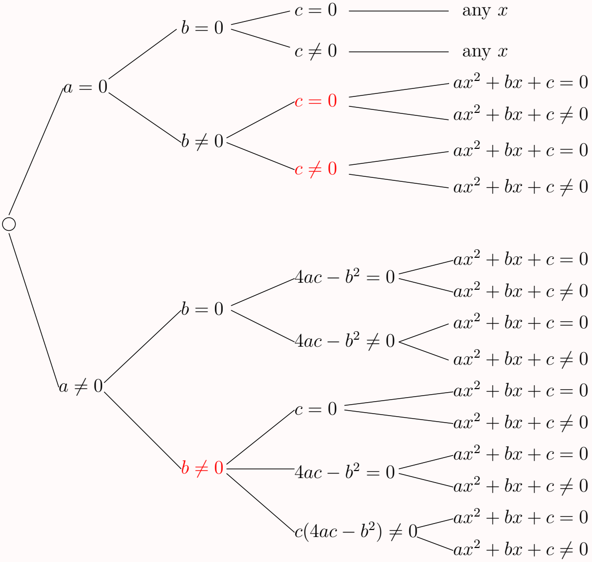
- for each $p \in P$, $\Phi_\alpha(\text{leading coefficient of p w.r.t. } x_n) \neq 0$
- the polynomials $\Phi_\alpha(p)$ are squarefree and pairwise coprime.

A $\{y^2 + x, y^2 + y\}$-sign invariant complex cylindrical tree

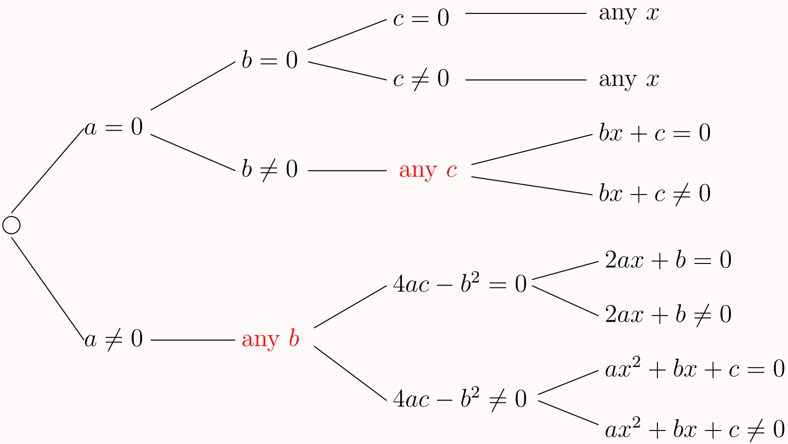


Rethink classical CAD in terms of complex cylindrical tree

The projection factors are $a, b, c, 4ac - b^2, ax^2 + bx + c$.



The complex cylindrical tree constructed by TCAD

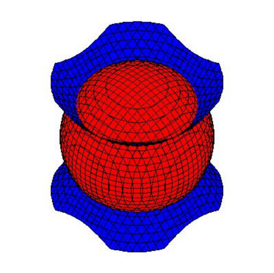


Outline

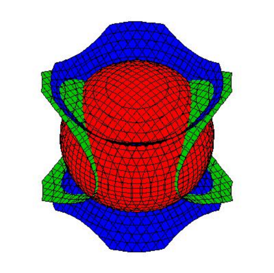
Incremental solving

- $p_1 := x^2 + y^2 + z^2 - 4$
- $p_2 := x^2 + y^2 - z^2 - 1$
- $p_3 := z^3 + xy - 1$

$W(T) := V(p_1) \cap V(p_2)$



$V(p_3) \cap W(T)$



Algorithms using incremental strategy: Triangular Decomposition (D. Lazard, 91; M^3, 00; C. Chen & M^3, 11); Lifting Fibers (G. Lecerf, 2003); Diagonal Homotopy (A.J. Sommese, J. Verschelde, C. W. Wampler, 08).

The refinement operation

Input

- A $y^2 + x$ sign invariant complex cylindrical tree

$$T := \begin{cases} x = 0 & \begin{cases} y = 0 & : \quad y^2 + x = 0 \\ y \neq 0 & : \quad y^2 + x \neq 0 \end{cases} \\ \\ x \neq 0 & \begin{cases} y^2 + x = 0 & : \quad y^2 + x = 0 \\ y^2 + x \neq 0 & : \quad y^2 + x \neq 0 \end{cases} \end{cases}$$

- A polynomial $y^2 + y$.

Output

The tree T is refined into a new tree, above each path of which both $y^2 + x$ and $y^2 + y$ are sign invariant.

Refine the first path of the tree with $y^2 + y$

$$\begin{cases} x = 0 & \begin{cases} y = 0 & : \quad y^2 + x = 0 \\ y \neq 0 & : \quad y^2 + x \neq 0 \end{cases} \\ \\ x \neq 0 & \cdots \end{cases}$$

$$\Rightarrow \quad \begin{cases} x = 0 & \begin{cases} y = 0 & : \quad y^2 + x = 0 \wedge y^2 + y = 0 \\ y \neq 0 & : \quad y^2 + x \neq 0 \end{cases} \\ \\ x \neq 0 & \cdots \end{cases}$$

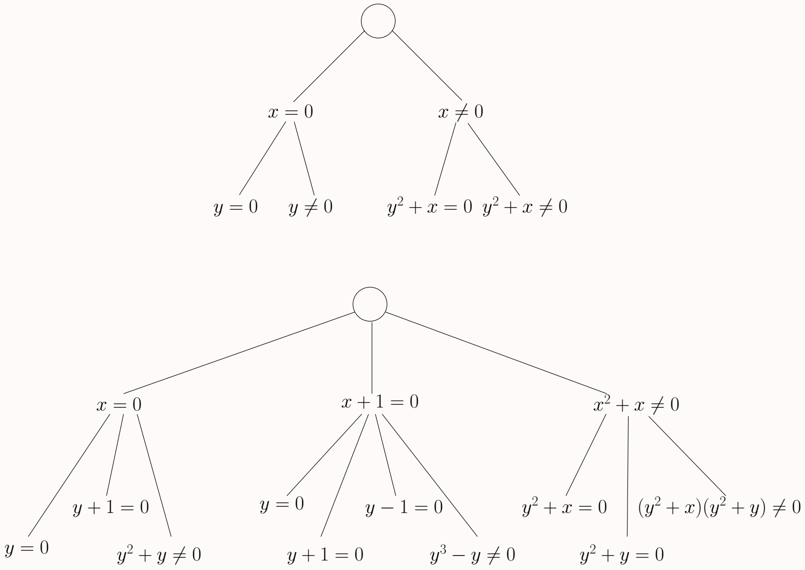
Refine the next path of the tree with y^2+y

$$\begin{cases} x=0 & \begin{cases} y=0 & : \quad y^2+x=0 \wedge y^2+y=0 \\ y\neq 0 & : \quad y^2+x\neq 0 \end{cases} \\ & \\ x\neq 0 & \cdots \end{cases}$$

$$\Rightarrow \quad \begin{cases} x=0 & \begin{cases} y=0 & : \quad y^2+x=0 \wedge y^2+y=0 \\ y=-1 & : \quad y^2+x\neq 0 \wedge y^2+y=0 \\ \text{otherwise} & : \quad y^2+x\neq 0 \wedge y^2+y\neq 0 \end{cases} \\ & \\ x\neq 0 & \cdots \end{cases}$$

The $\{y^2+x, y^2+y\}$ sign invariant cylindrical tree of $\mathbb{C}^2$

$$
\begin{cases}
x = 0 & \begin{cases} y = 0 & : \quad y^2+x=0 \wedge y^2+y=0 \\ y=-1 & : \quad y^2+x \neq 0 \wedge y^2+y=0 \\ \text{otherwise} & : \quad y^2+x\neq 0 \wedge y^2+y \neq 0 \end{cases} \\
x = -1 & \begin{cases} y = -1 & : \quad y^2+x=0 \wedge y^2+y=0 \\ y=1 & : \quad y^2+x = 0 \wedge y^2+y\neq 0 \\ y=0 & : \quad y^2+x \neq 0 \wedge y^2+y=0 \\ \text{otherwise} & : \quad y^2+x\neq 0 \wedge y^2+y \neq 0 \end{cases} \\
\text{otherwise} & \begin{cases} y^2+x = 0 & : \quad y^2+x=0 \wedge y^2+y\neq 0 \\ y^2+y=0 & : \quad y^2+x \neq 0 \wedge y^2+y=0 \\ \text{otherwise} & : \quad y^2+x\neq 0 \wedge y^2+y \neq 0 \end{cases}
\end{cases}
$$

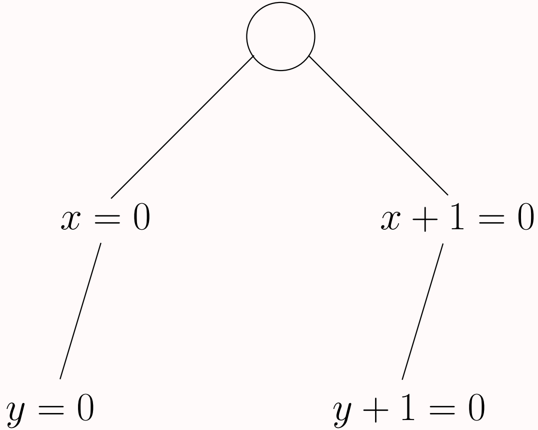


$x = 0$
$x \neq 0$
$y = 0$
$y \neq 0$
$y^2 + x = 0$
$y^2 + x \neq 0$
$x = 0$
$x + 1 = 0$
$x^2 + x \neq 0$
$y + 1 = 0$
$y = 0$
$y^2 + y \neq 0$
$y = 0$
$y - 1 = 0$
$y + 1 = 0$
$y^3 - y \neq 0$
$y^2 + x = 0$
$(y^2 + x)(y^2 + y) \neq 0$
$y^2 + y = 0$

Outline

Compute partial cylindrical tree

A partial cylindrical tree induced by the $F := \{y^2 + x = 0, y^2 + y = 0\}$ is



Transform a complex cylindrical decomposition to a real one

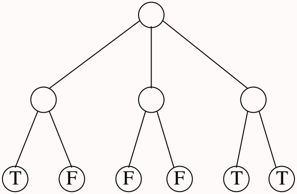
$$\text{Complex}: \begin{cases} x = 0 & \begin{cases} y = 0 & : \quad y^2 + x = 0 \\ y \neq 0 & : \quad y^2 + x \neq 0 \end{cases} \\ \\ x \neq 0 & \begin{cases} y^2 + x = 0 & : \quad y^2 + x = 0 \\ y^2 + x \neq 0 & : \quad y^2 + x \neq 0 \end{cases} \end{cases}$$

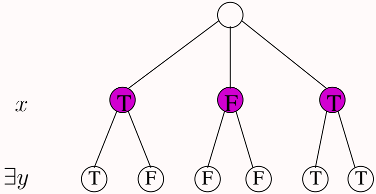
$$\text{Real}: \begin{cases} x < 0 & \begin{cases} y < -\sqrt{|x|} & : \quad y^2 + x > 0 \\ y = -\sqrt{|x|} & : \quad y^2 + x = 0 \\ y > -\sqrt{|x|} \wedge y < \sqrt{|x|} & : \quad y^2 + x < 0 \\ y = \sqrt{|x|} & : \quad y^2 + x = 0 \\ y > \sqrt{|x|} & : \quad y^2 + x > 0 \end{cases} \\ \\ x = 0 & \begin{cases} y < 0 & : \quad y^2 + x > 0 \\ y = 0 & : \quad y^2 + x = 0 \\ y > 0 & : \quad y^2 + x > 0 \end{cases} \\ \\ x > 0 & \text{for any } y : \quad y^2 + x > 0 \end{cases}$$

Outline

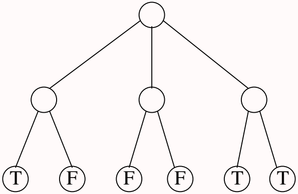
Propagate the truth values

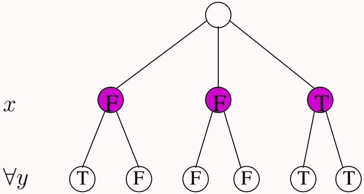
$(\exists y) f(x, y) \geq 0$



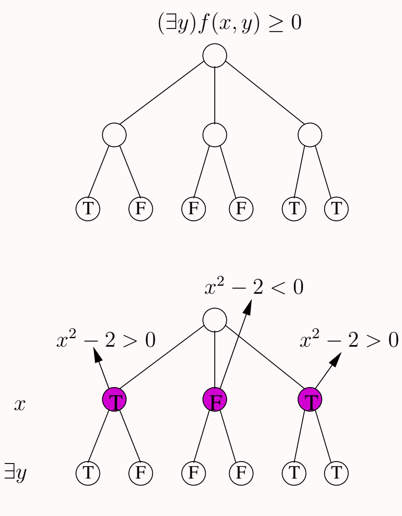


$(\forall y) f(x, y) \geq 0$





Generate the solution formula

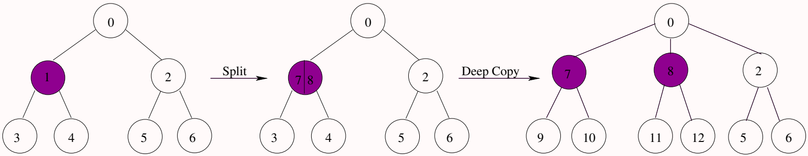


For the right example, to distinguish the true and false cells, one refines the tree w.r.t. the derivative of $x^2 - 2$. In general, one applies Thom's lemma.

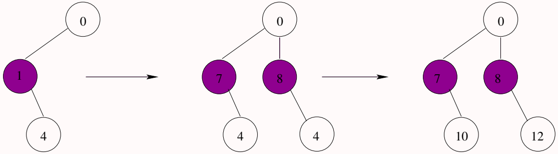
Outline

Implementation in Maple

The universe tree is always up-to-date



A sub-tree evolves with the universe tree

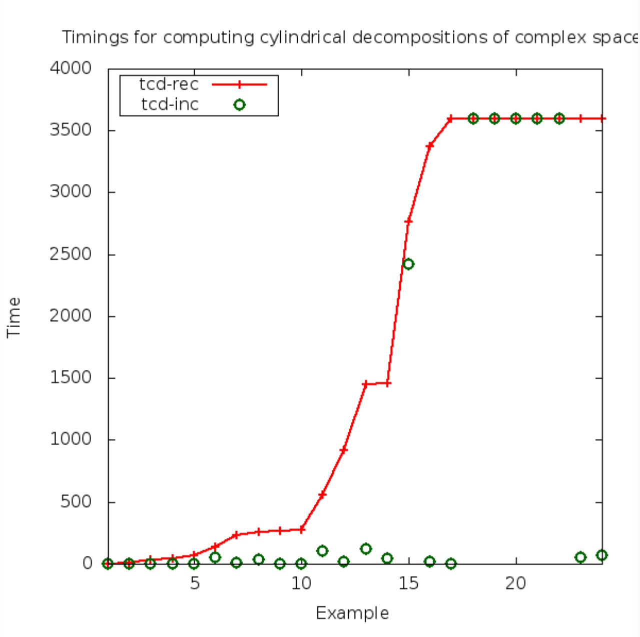


Information of the test examples

System	nvars	degree	nsys	nterms
AlkashiSinus	9	3	6	21
Alonso	7	4	4	13
Arnborg-Lazard-rev	3	6	3	21
Barry	3	5	3	9
blood-coagulation-2	4	4	3	15
Bronstein-Wang	4	4	3	22
cdc2-cyclin	3	9	2	8
circles	2	10	2	72
genLinSyst-3-2	11	2	3	9
GonzalezGonzalez	3	3	3	10
lhlp2	3	9	3	16
lhlp5	3	3	3	22
MontesS10	7	3	4	14
MontesS12	8	2	4	12
MontesS15	12	2	7	19
MontesS4	4	2	4	11
MontesS5	8	3	4	19
MontesS7	4	3	3	12
MontesS9	6	2	3	12
nql-5-4	5	4	5	14
r-5	5	6	5	18
r-6	6	7	6	22
Rose	3	9	3	29
Wang93	5	3	4	15

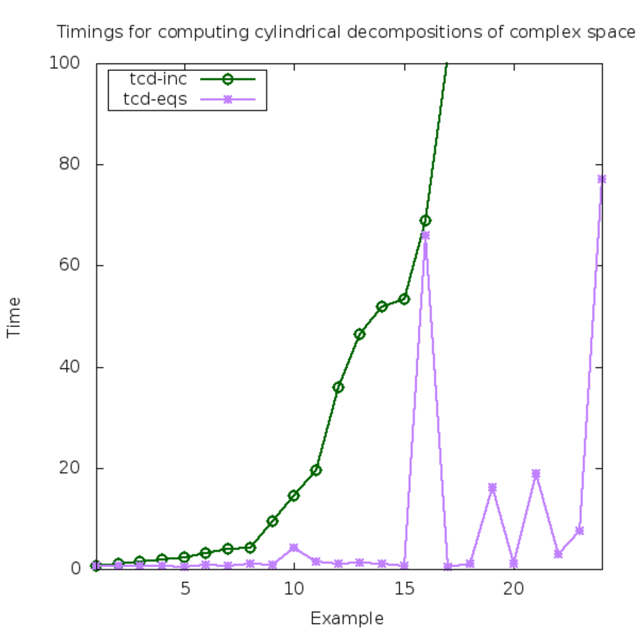
tcd-rec : our ISSAC'09 recursive algorithm

tcd-inc : the incremental algorithm



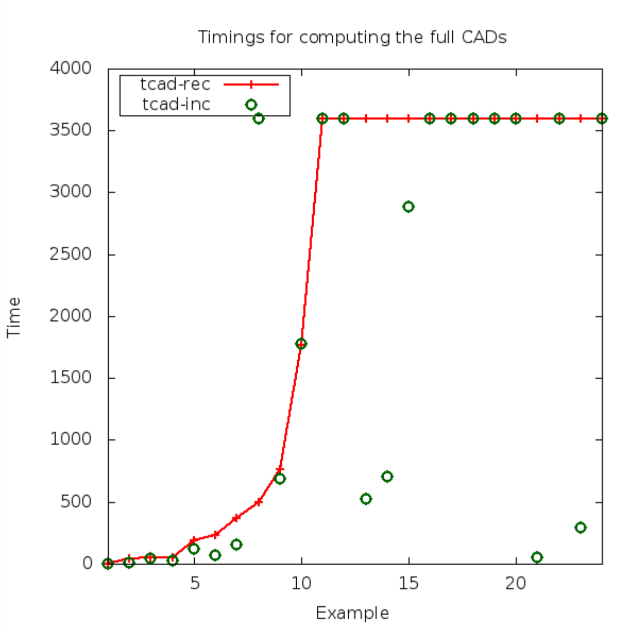
tcd-inc: the incremental algorithm with set of polynomials as input

tcd-eqs: the incremental algorithm with set of equations as input



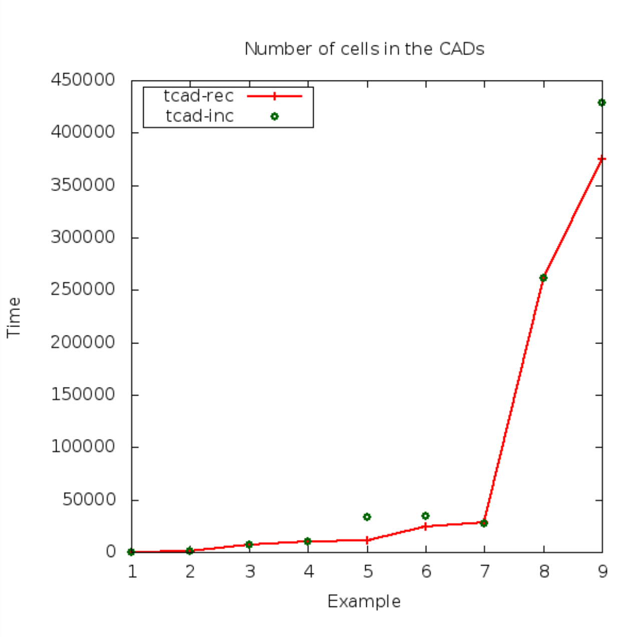
tcad-rec : our ISSAC'09 recursive algorithm

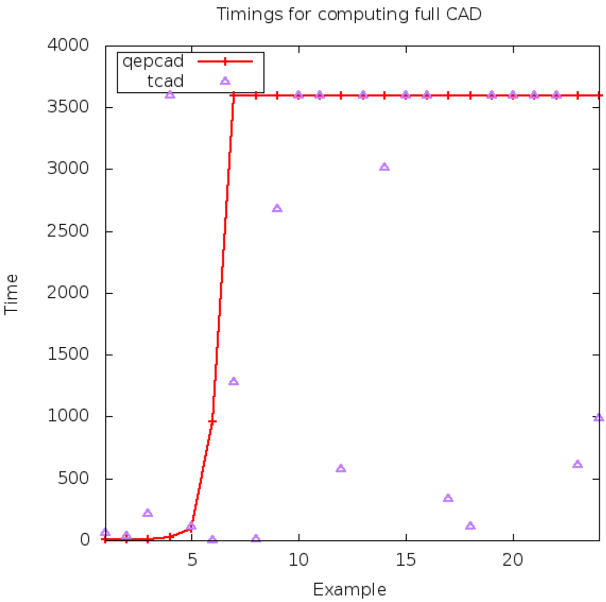
tcad-inc : the incremental algorithm

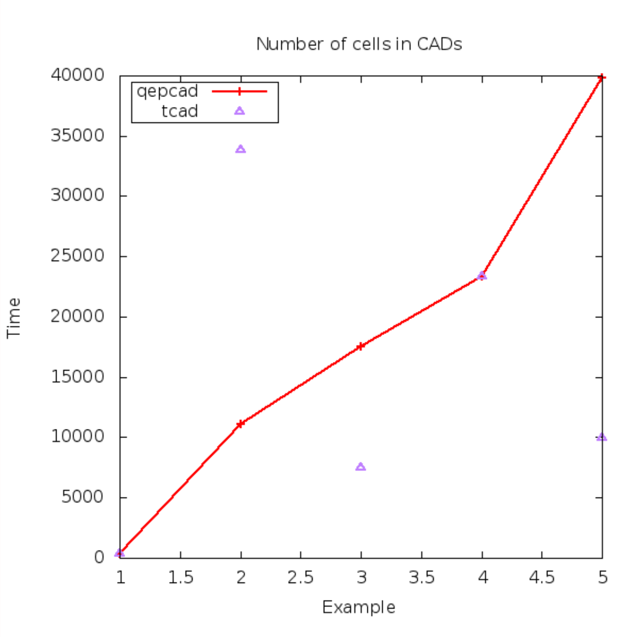


tcad-rec : our ISSAC'09 recursive algorithm

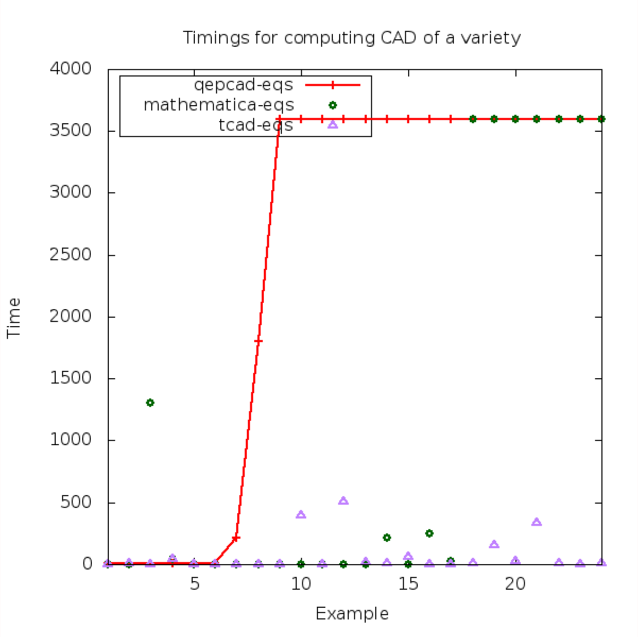
tcad-inc : the incremental algorithm




Timings for computing full CAD
qepcad
tcad
Time
Example
4000
3500
3000
2500
2000
1500
1000
500
0
5
10
15
20



Number of cells in CADs
qepcad
tcad
Time
Example
40000
35000
30000
25000
20000
15000
10000
5000
0
1
1.5
2
2.5
3
3.5
4
4.5
5

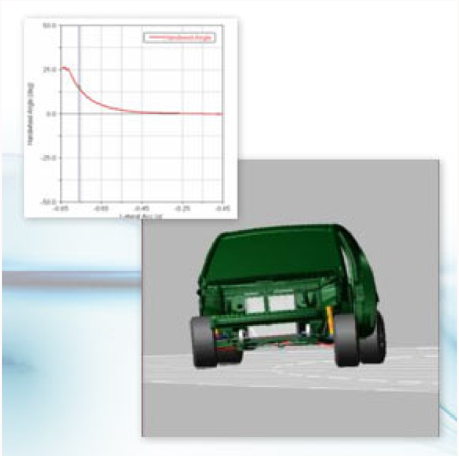


Timings for computing CAD of a variety
qepcad-eqs
mathematica-eqs
tcad-eqs
Time
Example
4000
3500
3000
2500
2000
1500
1000
500
0
5
10
15
20

Outline

1. First Idea: Introduce Case Discussion
2. Second Idea: Compute the Decomposition Incrementally
3. Third Idea: Compute CAD of a Variety
4. QE via TCAD
5. Implementation and Benchmark
6. Application

Stability of control system



www.maplesoft.com/industries/automotive/vehicle_dynamics

Control Lyapunov function

The control system

- $\dot{x} = f(x, u)$,
- $x \in \mathbb{R}^n$,
- u implicitly depends on x and t.

Control Lyapunov function

A function $V(x) : \mathbb{R}^n \to \mathbb{R}$ satisfying

- $V(x)$ is positive definite, that is $V(0) = 0$ and $\forall x \neq 0$, $V(x) > 0$.
- $\dot{V}(0) = 0$ and $\forall x \neq 0$, $\exists u$, such that $\dot{V} < 0$, where $\dot{V} = \frac{\partial V}{\partial x} \cdot f(x, u)$.
- V is radially unbounded, that is $\|x\| \to \infty$ implies that $V \to \infty$.

Find control Lyapunov function by QeTcad

Consider a dynamical system with variables x, y and a control parameter u.
We'd like to know if there exists a control Lyapunov function of the form $ax^2 + by^2$.
The equivalent QE problem is: $\forall\,(x, y)\,\exists\,(u)\left(x \neq 0 \textbf{ or } y \neq 0 \Rightarrow V > 0 \wedge \frac{d}{dt} V < 0\right)$.

The input control system.

```
> S := {diff(x(t), t) = -x(t)+u, diff(y(t),t) = -x(t) - y(t)^3};
```

$$S := \left\{\frac{d}{dt}x(t) = -x(t) + u, \frac{d}{dt}y(t) = -x(t) - y(t)^3\right\}$$

Compute $\frac{d}{dt}V$.

```
> fx := -x+u; fy := -x-y^3; V := a*x^2+b*y^2;
  Vt := diff(V, x)*fx + diff(V, y)*fy;
```

$$fx := -x + u$$

$$fy := -y^3 - x$$

$$V := a\,x^2 + b\,y^2$$

$$Vt := 2\,a\,x\,(-x + u) + 2\,b\,y\,(-y^3 - x)$$

Call QeTcad to find conditions on a and b.

```
> QuantifierElimination(&A([x,y]), &E([u]), (x<>0) &or (y<>0) &implies ((V>0) &and (Vt<0)));
```

$$0 < b \,\&\text{and}\; 0 < a$$

Verify the result for $a = 1$, $b = 4$.
Indeed $x^2 + 4\,y^2$ is a control Lyapunov function.

```
> simplify(subs({a=1,b=4}, Vt)); subs(u=4*y, %);
```

$$-8\,y^4 + 2\,u\,x - 2\,x^2 - 8\,x\,y$$

$$-8\,y^4 - 2\,x^2$$

Conclusion and work in progress

Conclusion

- We presented an incremental algorithm for computing CADs.
- The core operation of our algorithm is an Intersect operation, which refines a complex cylindrical tree by means of a polynomial constraint.
- The Intersect operation provides a systematic solution for propagating equational constraints.
- For many examples, the incremental outperforms both QEPCAD and Mathematica as well as our previous recursive algorithm.
- We have developed a QE routine QETCAD based on TCAD.

Work in progress

- We are working on different optimizations for both TCAD and QETCAD.

Table: Timings for computing CAD

System	nvars	degree	nsys	nterms	qepcad	qepcad-eqs	math-eqs	tcad	tcad-eqs
AlkashiSinus	9	3	6	21	$> 1h$	$> 1h$	2.232	$> 1h$	58.775
Alonso	7	4	4	13	7.516	5.284	0.74	61.591	5.776
Arnborg-Lazard-rev	3	6	3	21	$> 1h$	$> 1h$	0.952	$> 1h$	17.325
Barry	3	5	3	9	Fail	216.425	0.032	8.580	1.004
blood-coagulation-2	4	4	3	15	$> 1h$	$> 1h$	$> 1h$	985.709	7.260
Bronstein-Wang	4	4	3	22	$> 1h$	$> 1h$	26.726	333.892	2.564
cdc2-cyclin	3	9	2	8	$> 1h$	$> 1h$	0.208	574.127	503.863
circles	2	10	2	72	21.633	5.996	41.211	$> 1h$	40.902
genLinSyst-3-2	11	2	3	9	Fail	Fail	217.062	3013.764	6.588
GonzalezGonzalez	3	3	3	10	10.528	10.412	0.012	214.213	1.136
lhlp2	3	9	3	16	960.756	5.076	0.016	3.124	0.952
lhlp5	3	3	3	22	10.300	10.068	0.016	35.338	1.084
MontesS10	7	3	4	14	$> 1h$	$> 1h$	$> 1h$	$> 1h$	22.797
MontesS12	8	2	4	12	$> 1h$	$> 1h$	$> 1h$	$> 1h$	330.996
MontesS15	12	2	7	19	$> 1h$	$> 1h$	0.004	$> 1h$	395.964
MontesS4	4	2	4	11	$> 1h$	$> 1h$	0.004	2682.391	0.888
MontesS5	8	3	4	19	Fail	Fail	$> 1h$	$> 1h$	9.400
MontesS7	4	3	3	12	$> 1h$	$> 1h$	245.807	$> 1h$	2.452
MontesS9	6	2	3	12	Fail	Fail	$> 1h$	110.902	4.944
nql-5-4	5	4	5	14	93.073	5.420	1303.07	113.675	1.004
r-5	5	6	5	18	$> 1h$	1802.676	0.016	1282.928	1.208
r-6	6	7	6	22	$> 1h$	$> 1h$	0.024	$> 1h$	1.500
Rose	3	9	3	29	Fail	$> 1h$	$> 1h$	606.361	3.136
Wang93	5	3	4	15	Fail	Fail	$> 1h$	$> 1h$	152.673