REAL SOLVING BIVARIATE POLYNOMIAL SYSTEMS

theory and maple implementation

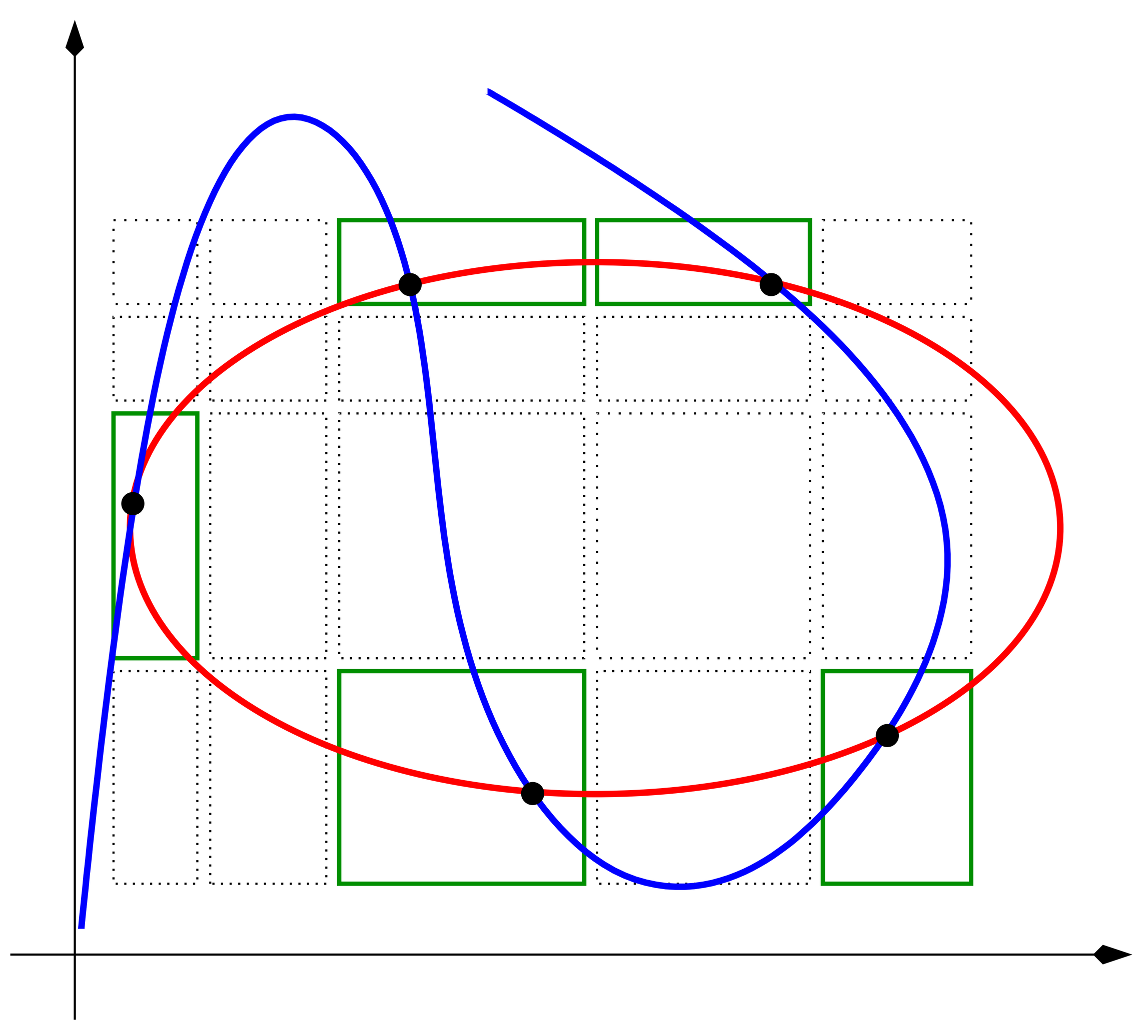
Dimitrios I. Diochnos, Ioannis Z. Emiris and Elias P. Tsigaridas

d.diochnos(AT)di.uoa.gr, emiris(AT)di.uoa.gr, elias.tsigaridas(AT)loria.fr

Bivariate Real Solving

We consider the following system, where $F, G \in \mathbb{Z}[x, y]$ and the maximum coefficient bitsize is τ. Let $N = \max\{d, \tau\}$.

$$\begin{cases} F = \sum_{1 \le i \le d} \sum_{1 \le j \le d} a_{i,j}\, x^i y^j = 0 \\ G = \sum_{1 \le i \le d} \sum_{1 \le j \le d} b_{i,j}\, x^i y^j = 0 \end{cases} \qquad (1)$$



Previous bounds. Bounds for bivariate real solving come from the algorithms for computing the topology of real plane algebraic curves.

- Using isolating intervals: $\widetilde{\mathcal{O}}_B(N^{30})$ [1]
- Using Thom's encoding: $\widetilde{\mathcal{O}}_B(N^{14})$ [2, 4]

Useful results

Lemma. [Aggregate separation] Given $f \in \mathbb{Z}[x]$ of degree d and bitsize τ, the sum of the bitsize of *all* the isolating points of the real roots of f is $\mathcal{O}(d^2 + d\,\tau)$.

Let $f, g \in (\mathbb{Z}[y_1, \ldots, y_k])[x]$ with $dg_x(f) = p \ge q = dg_x(g)$, $dg_{y_i}(f) \le d_i$ and $dg_{y_i}(g) \le d_i$. Let $d = \prod_{i=1}^{k} d_i$, τ be the bitsize of f and g and $\mathbf{SR}(f, g)$ be the Sylvester-Habicht sequence of f and g, w.r.t. x.

Proposition.

1. We can compute any polynomial in $\mathbf{SR}(f, g)$, and $\mathtt{res}(f, g)$ in $\widetilde{\mathcal{O}}_B(q(p+q)^{k+1} d\tau)$.
2. $\mathbf{SR}(f, g)$ is computed in $\widetilde{\mathcal{O}}_B(q(p+q)^{k+2} d\tau)$.
3. We can evaluate $\mathbf{SR}(f, g)$ at $x = \alpha$, where $a \in \mathbb{Q} \cup \{\infty\}$ and has bitsize σ in $\widetilde{\mathcal{O}}_B(q(p+q)^{k+1} d \max\{\tau, \sigma\})$.

The algorithms

The main steps of all the algorithms are the following:

(i) Project on the x-axis and isolate the real roots of the resultant.
(ii) Project on the y-axis and isolate the real roots of the resultant.
(iii) *Match* the solutions!
In the figure we have to identify the green boxes.

The algorithms differ in the way they match the solutions.

- GRID (no assumptions)
  Match the solutions using bivariate sign evaluations. For every real root on the x–axis, say α, for every real root on the y–axis, say β, test if $\operatorname{sign}(F(\alpha, \beta)) = \operatorname{sign}(G(\alpha, \beta)) = 0$.
  Complexity: $\widetilde{\mathcal{O}}_B(N^{14})$.
- M_RUR (assume generic position)
  For every real root on the x–axis, say α, using the properties of the Sturm-Habicht sequences express the ***unique*** y-coordinate as rational univariate function (RUR) of α. Finally, match the RUR with the corresponding β, using univariate sign evaluations.
  Complexity: $\widetilde{\mathcal{O}}_B(N^{12})$.
- G_RUR (no assumptions)
  For every real root on the x–axis, say α, compute the square-free parts $F_{red}, G_{red} \in (\mathbb{Z}[\alpha])[y]$. Compute $H = \gcd(F_{red}, G_{red}) \in (\mathbb{Z}[\alpha])[y]$. Check for solutions of $H_{red}(\alpha, y)$ among the y–intervals (must change sign).
  Complexity: $\widetilde{\mathcal{O}}_B(N^{12})$.

Theorem. We can compute the pairs of real algebraic numbers that are solutions of (1) (and the multiplicities) in $\widetilde{\mathcal{O}}_B(N^{12})$.

Corollary. We can compute the topology of a real plane algebraic curve in $\widetilde{\mathcal{O}}_B(N^{12})$.

For further details you may refer to [3].

SLV: a MAPLE library

SLV stands for **S**turm-based so**LV**er. It is a MAPLE library for computations with real algebraic numbers. It currently computes the real solutions in isolating interval representation, for univariate polynomials or well constrained systems of bivariate polynomials and comparison and sign evaluations with one and two real algebraic numbers. The library uses filtering techniques to speed up the computations.
SLV is available at `www.di.uoa.gr/~erga/soft/SLV_index.html`

Univariate real solving

Using the library we isolate the real roots of an integer univariate polynomial and thus construct real algebraic numbers in isolating interval representation.

```
> LIBPATH := "/opt/SLV/":
> read cat ( LIBPATH, "system.mpl" ):
> f := 3*x^3 - x^2 - 6*x + 2:
> sols := SLV:-solveUnivariate( f ):
> SLV:-display_1 ( sols );

   < x^2-2, [ -93/64, -45/32], -1.414213568 >
   < 3*x-1, [ 1/3, 1/3], 1/3 >
   < x^2-2, [ 45/32, 93/64], 1.414213568 >
```

Bivariate real solving

The library provides three algorithms for real solving a bivariate polynomial system, namely GRID, M_RUR and G_RUR. The output is a list of pairs of real algebraic numbers.

```
> f := 1+2*x+x^2*y-5*x*y+x^2:
> g := 2*x+y-3:
> bivsols := SLV:-solveGRID ( f, g ):
> SLV:-display_2 ( bivsols );

   < 2*x^2-12*x+1, [ 3, 7], 5.915475965 > ,
       < x^2+6*x-25, [ -2263/256, -35/4], -8.830718995 >
   < x-1, [ 1, 1], 1 > , < x-1, [ 1, 1], 1 >
   < 2*x^2-12*x+1, [ 3/64, 3/32], .8452400565e-1 >,
       < x^2+6*x-25, [ 23179/8192, 2899/1024], 2.830943108 >
```

We can also use the functions `SLV:-solveGRID` or `SLV:-solveMRUR`.

Operation with real algebraic numbers

Currently the library provides functions for comparing two real algebraic numbers and for computing the sign of a polynomial if we evaluate it over one or two real algebraic numbers.

```
> alpha := SLV:-solve_1( x^4 - 7)[2]:  alpha:-display();
> alpha := SLV:-solve_1( x^6 - 13)[2]: beta:-display();

   < x^4-7, [ 1, 2], 1.626556396 >
   < x^6-13, [ 1, 2], 1.533386230 >

> FK:-compare( alpha, alpha);               0
> FK:-compare( alpha, beta);                1
> FK:-compare( beta,  alpha);              -1

> FK:-signAt( x^4 - 7, alpha);              0
> FK:-signAt( x^2 - y^2 - 2, alpha, beta);  -1
```

Experiments on bivariate real solving

system	deg		solutions	Average Time (msecs)									
				BIVARIATE SOLVING							TOPOLOGY		
				this paper (SLV)			FGb/Rs	Synaps			Insulate	Top	
	f	g		grid	m_rur	g_rur		sturm	subdiv	newmac		60	500
R_1	3	4	2	5	9	5	26	2	2	5	–	–	–
R_2	3	1	1	66	21	36	24	1	1	1	–	–	–
R_3	3	1	1	1	2	1	22	1	2	1	–	–	–
M_1	3	3	4	87	72	10	25	2	1	2	–	–	–
M_2	4	2	3	4	5	4	24	1	289	2	–	–	–
M_3	6	3	5	803	782	110	30	230	5,058	7	–	–	–
M_4	9	10	2	218	389	210	158	90	3	447	–	–	–
D_1	4	5	1	6	12	6	28	2	5	8	–	–	–
D_2	2	2	4	667	147	128	26	21	1	2	–	–	–
C_1	7	6	6	1,896	954	222	93	479	170,265	39	524	409	1,367
C_2	4	3	6	177	234	18	27	12	23	4	28	36	115
C_3	8	7	13	580	1,815	75	54	23	214	25	327	693	2,829
C_4	8	7	17	5,903	80,650	370	138	3,495	217	190	1,589	1,624	6,435
C_5	16	15	17	$> 20'$	60,832	3,877	4,044	$> 20'$	6,345	346	179,182	91,993	180,917
W_1	7	6	9	2,293	2,115	247	92	954	55,040	39	517	419	1,350
W_2	4	3	5	367	283	114	29	20	224	3	27	20	60
W_3	8	7	13	518	2,333	24	56	32	285	25	309	525	1,588
W_4	8	7	17	5,410	77,207	280	148	4,086	280	207	1,579	1,458	4,830

GRID: Matching (bivariate SignAt) is the bottleneck (73%)
Filtering speedup up to 10×

M_RUR: Projection = 13%, Matching=45–50%
Filtering speedup up to 10×

G_RUR: Projection=29%, Matching (including GCD) = 70%
Filtering speedup up to 2×

Overall evaluation

- G_RUR up to 11 times faster than GRID
- G_RUR up to 38 times faster than M_RUR

References

[1] D. Arnon and S. McCallum. A polynomial time algorithm for the topological type of a real algebraic curve. *JSC*, 5:213–236, 1988.

[2] S. Basu, R. Pollack, and M-F.Roy. *Algorithms in Real Algebraic Geometry*, volume 10 of *Algorithms and Computation in Mathematics*. Springer-Verlag, 2nd edition, 2006.

[3] D. I. Diochnos, I. Z. Emiris, and E. P. Tsigaridas. On the complexity of real solving bivariate systems. In C. W. Brown, editor, *Proc. Annual ACM Intern. Symp. on Symbolic and Algebraic Computation (ISSAC)*, Waterloo, Canada, 2007. Also available as INRIA RR 6116, https://hal.inria.fr/inria-00129309.

[4] L. González-Vega and M. El Kahoui. An improved upper complexity bound for the topology computation of a real algebraic plane curve. *J. Complexity*, 12(4):527–544, 1996.