Applications of Real Algebraic Varieties

Daniel Brake

14 March, 2015

Joint work with
Jon Hauenstein
and
Cynthia Vinzant

UNIVERSITY OF NOTRE DAME
College of Science

Tropicalization

The tropicalization of an equidimensional algebraic variety is typically defined for complex components.

- For complex plane curves, can be defined as limits of 'amoebas':

$$\lim_{t\to 0}\left(-\frac{\log|z_1|}{\log t}, -\frac{\log|z_2|}{\log t}\right)$$

- Can also be defined in terms of Puiseux series.
- Can also be defined in terms of the max-plus semiring

Gathmann. Tropical Algebraic Geometry. arXiv:math/0601322v1

Amoeba

The complex amoeba of

$$P(z,w) = 50z^3+83z^2w+24zw^2+w^3+392z^2+414zw+50w^2-28z+59w-100$$

Existing methods for computing Tropical Curves

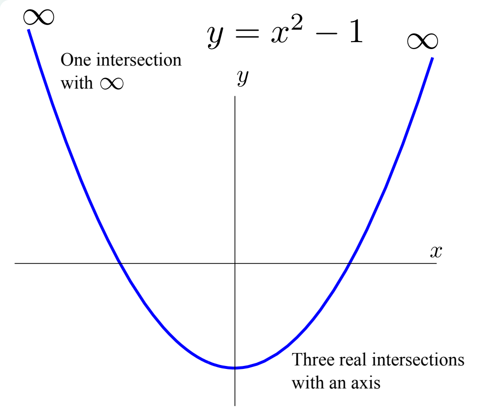
There are implemented ways to compute $\mathbb{C}$ tropical curves:

- `Gfan`, Gröbner bases.
- Homotopy Continuation [JLY], amoebas.
  1. Find possible rays by sampling tentacles,
  2. Compute multiplicities of rays,
  3. Check for completeness.

Neither above method can compute the $\mathbb{R}$ tropicalization (or tropicalization of a $\mathbb{R}$ curve), so we will explore what this means, and how to compute it.

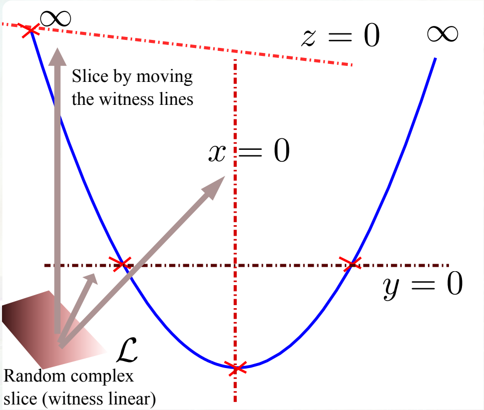
Jensen, Leykin, Yu. Computing Tropical Curves Via Homotopy Continuation. arXiv:1408.3105v1

A $\mathbb{R}$ example – a parabola



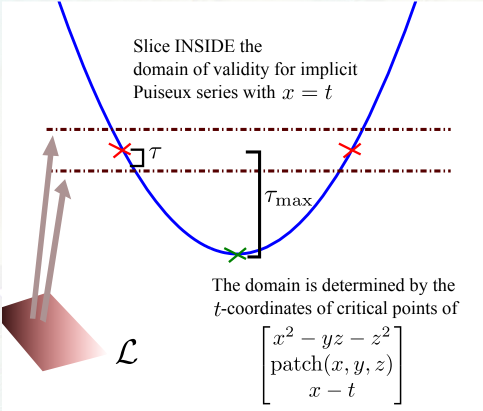
Projectivizing the parabola lets us compute the behaviour at ∞, when we supplement with a random real patch.

Parabola (ctd)



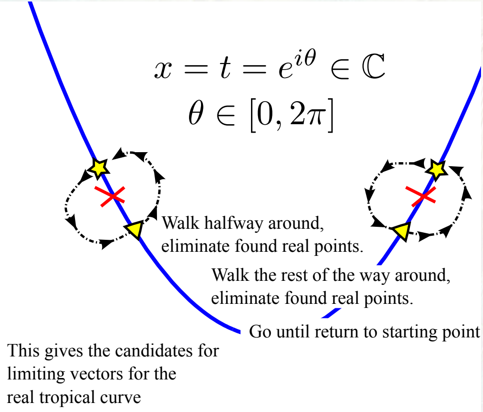
Slicing the curve at each coordinate tell us whether to do further computations on that coordinate.

Parabola (ctd)



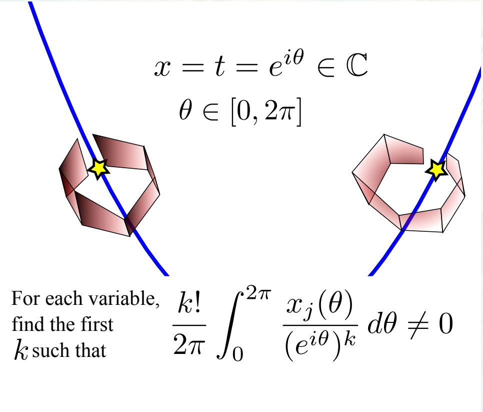
Get the $\mathbb{R}$ points for Puiseux series computation

Parabola (ctd)



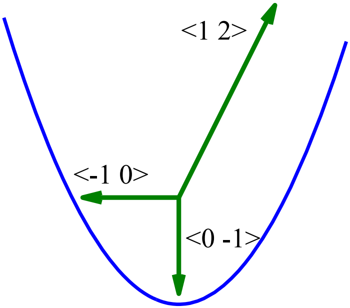
Doing monodromy loops determines the distinct paths limiting to the real intersection points.

Parabola (ctd)



Find the valuations for each variable, by estimating Cauchy integrals via the trapezoid rule.

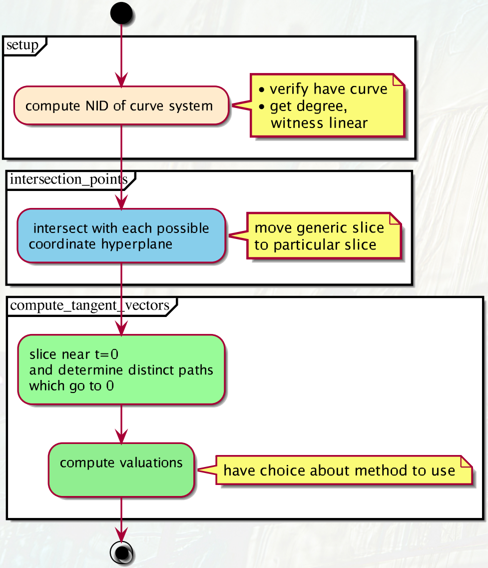
Parabola (ctd)

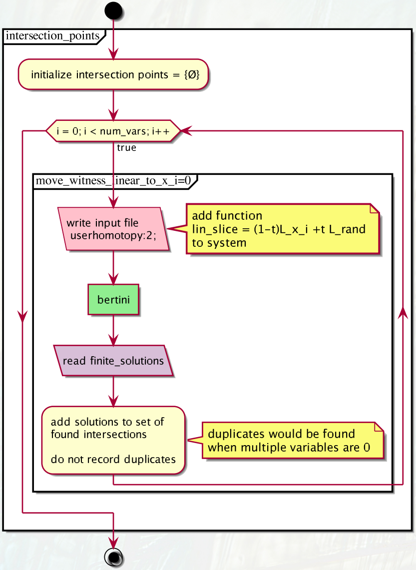


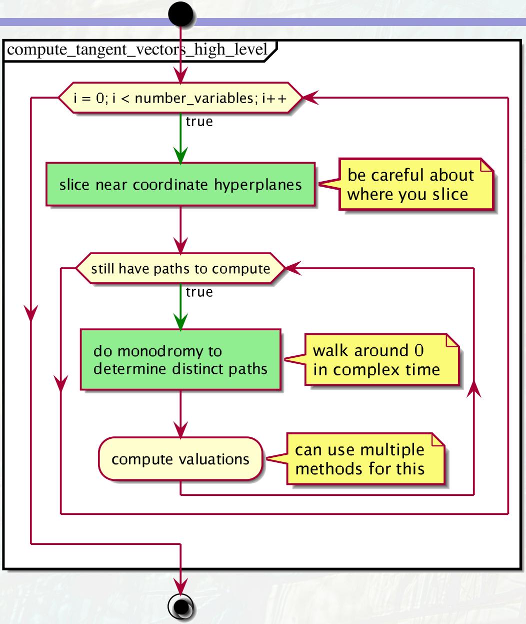
After dehomogenization, we get this picture
for the real tropicalization of this parabola

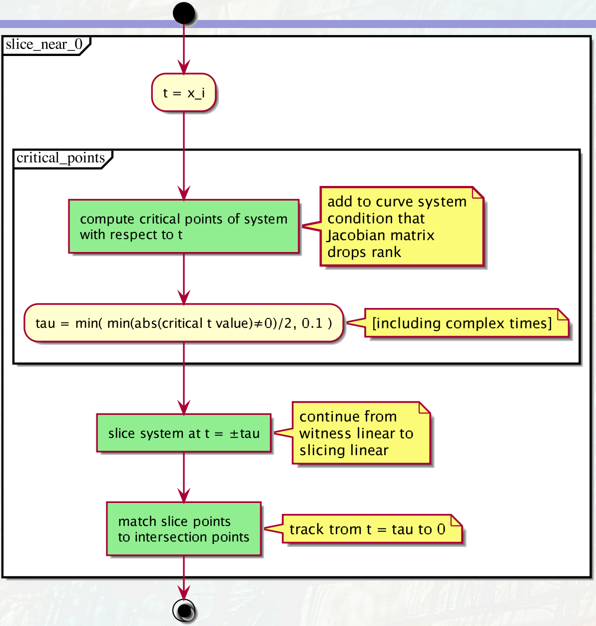
Repeating the steps, we get the real tropicalization of this parabola.

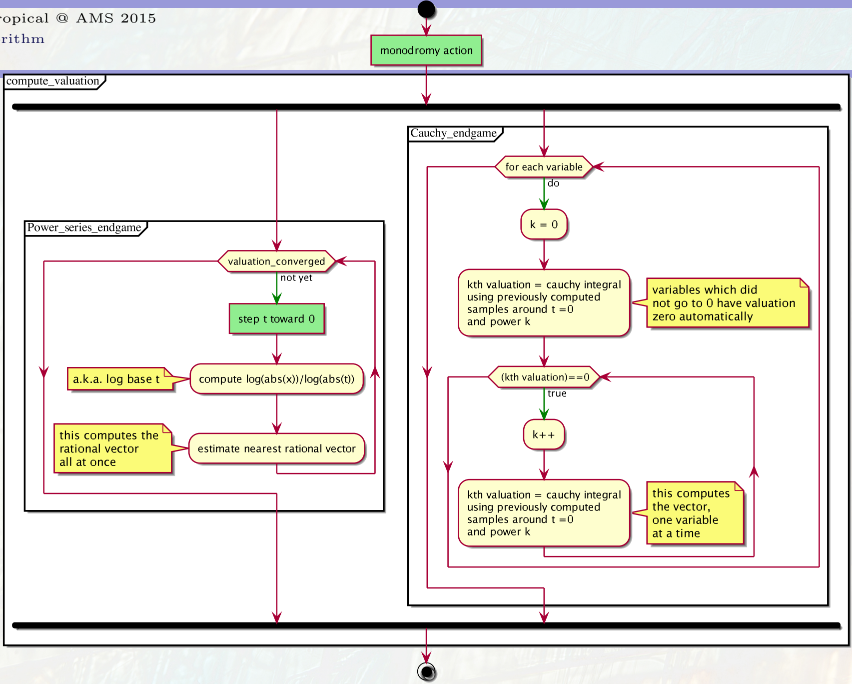
High level


setup
compute NID of curve system
• verify have curve
• get degree,
witness linear
intersection_points
intersect with each possible
coordinate hyperplane
move generic slice
to particular slice
compute_tangent_vectors
slice near t=0
and determine distinct paths
which go to 0
compute valuations
have choice about method to use

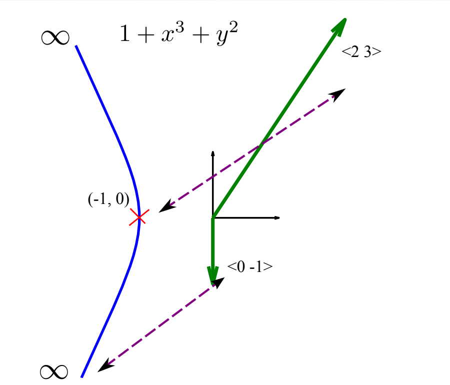

intersection_points
initialize intersection points = {Ø}
i = 0; i < num_vars; i++
true
move_witness_linear_to_x_i=0
write input file
userhomotopy:2;
add function
lin_slice = (1-t)L_x_i +t L_rand
to system
bertini
read finite_solutions
add solutions to set of
found intersections
do not record duplicates
duplicates would be found
when multiple variables are 0


compute_tangent_vectors_high_level
i = 0; i < number_variables; i++
true
slice near coordinate hyperplanes
be careful about where you slice
still have paths to compute
true
do monodromy to determine distinct paths
walk around 0 in complex time
compute valuations
can use multiple methods for this


slice_near_0
t = x_i
critical_points
compute critical points of system with respect to t
add to curve system condition that Jacobian matrix drops rank
tau = min(min(abs(critical t value)≠0)/2, 0.1)
[including complex times]
slice system at t = ±tau
continue from witness linear to slicing linear
match slice points to intersection points
track trom t = tau to 0

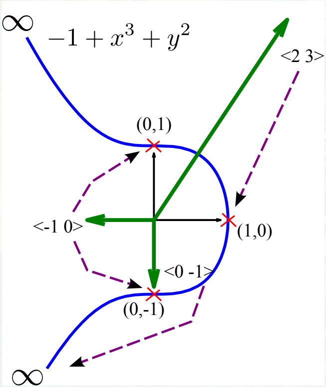

monodromy action
compute_valuation
Power_series_endgame
valuation_converged
not yet
step t toward 0
a.k.a. log base t
compute log(abs(x))/log(abs(t))
this computes the
rational vector
all at once
estimate nearest rational vector
Cauchy_endgame
for each variable
do
k = 0
kth valuation = cauchy integral
using previously computed
samples around t =0
and power k
variables which did
not go to 0 have valuation
zero automatically
(kth valuation)==0
true
k++
kth valuation = cauchy integral
using previously computed
samples around t =0
and power k
this computes
the vector,
one variable
at a time

Elliptic Curve



Sometimes the real curve does not generate all the vectors of the complex tropicalization.

Elliptic Curve 2



The values of coefficients and parameters matter greatly!

Notes

- The valuations can be computed with either
  - the limit definition,
  $$\lim_{t \to 0} \frac{|z(t)|}{|t|}$$
  - Cauchy integrals,
  $$C \int_{\gamma} \frac{z(t)}{t^k}\, dt$$
- The limit definition has slow convergence (experiments under way).
- Since monodromy loops are periodic, the trapezoid rule for Cauchy integration converges quickly.
- Since one determines which slice points connect at $\cap$ points, and one can use monodromy for this, the Cauchy valuation method is superior.

Contrasts between $\mathbb{R}$ and $\mathbb{C}$ Tropicalization

$\mathbb{C}$

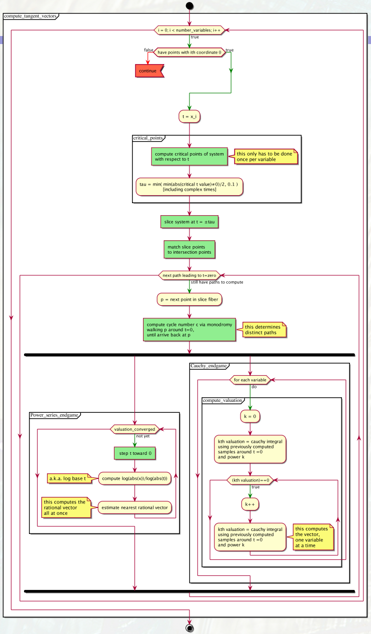
- Generic behaviour – largely doesn't depend on parameter values.
- Multiplicities can be seen as total # of Puiseux series at intersections with coordinate hyperplanes.
- Balancing condition to check for correct computation. Sum of all vectors $= 0$.

$\mathbb{R}$

- Non-generic behaviour – strongly depends on parameter values.
- Not clear yet how multiplicity is defined.
- No balancing condition. Unknown how to check for correct computation.

Thank you for your kind attention

Vectors


compute_tangent_vectors
i = 0; i < number_variables; i++
true
have points with ith coordinate 0
false
true
continue
t = x_i
critical_points
compute critical points of system
with respect to t
this only has to be done
once per variable
tau = min(min(abs(critical t value)≠0)/2, 0.1)
[including complex times]
slice system at t = ±tau
match slice points
to intersection points
next path leading to t=zero
still have paths to compute
p = next point in slice fiber
compute cycle number c via monodromy
walking p around t=0,
until arrive back at p
this determines
distinct paths
Power_series_endgame
valuation_converged
not yet
step t toward 0
a.k.a. log base t
compute log(abs(x))/log(abs(t))
this computes the
rational vector
all at once
estimate nearest rational vector
Cauchy_endgame
for each variable
do
compute_valuation
k = 0
kth valuation = cauchy integral
using previously computed
samples around t =0
and power k
(kth valuation)==0
true
k++
kth valuation = cauchy integral
using previously computed
samples around t =0
and power k
this computes
the vector,
one variable
at a time