Tropical Implicitization

Jan Verschelde

University of Illinois at Chicago
Department of Mathematics, Statistics, and Computer Science
`http://www.math.uic.edu/~jan`
`jan@math.uic.edu`

Graduate Computational Algebraic Geometry Seminar

Tropical Curves and Amoebas

1. Introduction
   - Introduction to Tropical Geometry
2. Implicitization
   - problem statement
   - solving the implicitization problem
3. Tropical Implicitization
   - computing the Newton polygon of an example
   - an algorithm to compute the Newton polygon

Tropical Curves and Amoebas

1. Introduction
   - Introduction to Tropical Geometry
2. Implicitization
   - problem statement
   - solving the implicitization problem
3. Tropical Implicitization
   - computing the Newton polygon of an example
   - an algorithm to compute the Newton polygon

Introduction to Tropical Geometry

Introduction to Tropical Geometry is the title of a forthcoming book of Diane Maclagan and Bernd Sturmfels.

The web page
`http://homepages.warwick.ac.uk/staff/D.Maclagan/papers/TropicalBook.html`
offers the pdf file of the first five chapters (23 August 2013).

Tropical islands is the title of the first chapter,
which promises a friendly welcome to tropical mathematics.

Today we look at section 1.5.

overview of the book

The titles of the five chapters with some important sections:

1. Tropical Islands
   - amoebas and their tentacles
   - implicitization
2. Building Blocks
   - polyhedral geometry
   - Gröbner bases
   - tropical bases
3. Tropical Varieties
   - the fundamental theorem
   - the structure theorem
   - multiplicities and balancing
   - connectivity and fans
   - stable intersection
4. Tropical Rain Forest
5. Linear Algebra

Tropical Curves and Amoebas

1. Introduction
   - Introduction to Tropical Geometry

2. Implicitization
   - problem statement
   - solving the implicitization problem

3. Tropical Implicitization
   - computing the Newton polygon of an example
   - an algorithm to compute the Newton polygon

the implicitization problem

Definition

An *unirational* algebraic variety can be represented

- either as the image of a rational map;
- or as the zero set of some multivariate polynomials.

Both representations have their specific applications, e.g.:

- The first representation as image of rational map is convenient for plotting the coordinates of the solutions.
- The second representation as the zero set of multivariate polynomials is needed for the ideal membership problem.

Definition

In Computer Algebra, *implicitization* is the problem of passing from the image as a rational map representation of a variety to the prime ideal of all polynomials that vanish on the image of the map.

using resultants in sympy

$$\Phi(t) = \left(\frac{t^3 + 4t^2 + 4t}{t^2 - 1}, \frac{t^3 - t^2 - t + 1}{t^2} \right).$$

Running the script

```
import sympy as sp
t, x, y = sp.var('t, x, y')
px = t**3 + 4*t**2 + 4*t - (t**2 - 1)*x
py = t**3 - t**2 - t + 1 - t**2*y
r = sp.resultant(px, py, t)
print r
```

produces

```
x**3*y**2 - x**2*y**3 - 5*x**2*y**2 - 2*x**2*y \
- 4*x*y**2 - 33*x*y + 16*y**2 + 72*y + 81
```

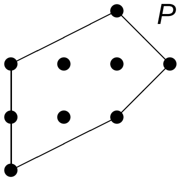
the Newton polygon

Definition

Given a polynomial $f(x,y) = c_{k,\ell} x^k y^\ell$, *the Newton polygon P of f* is the convex hull of the set $\{ \ (k,\ell) \ : \ c_{k,\ell} \neq 0 \ \}$.

Example:

$f(x,y) = x^3y^2 - x^2y^3 - 5x^2y^2 - 2x^2y - 4xy^2 - 33xy + 16y^2 + 72y + 81.$



Given the parametrization Φ and the Newton polygon P, what is f?

defining a linear system

A sympy script to setup a linear system:

```
import sympy as sp
from fractions import Fraction
t, x, y = sp.var('t, x, y')
fx = lambda t: Fraction(t**3 + 4*t**2 + 4*t)/(t**2 - 1)
fy = lambda t: Fraction(t**3 - t**2 - t + 1)/t**2
samples = []
for t in range(-5, -1):
    samples.append((fx(t), fy(t)))
for t in range(2, 6):
    samples.append((fx(t), fy(t)))
polygon = [(3, 2), (2, 3), (2, 2), (2, 1), (1, 2), \
           (1, 1), (0, 2), (0, 1), (0, 0)]
```

The list `samples` contains points for $t = \pm 2, \pm 3, \pm 4, \pm 5$:
eight points to determine nine coefficients.

the sympy script continued

```
L = []
for point in samples:
    (a, b) = (point[0], point[1])
    values = []
    for monomial in polygon:
        values.append(a**monomial[0]*b**monomial[1])
    L.append(values)
M = sp.Matrix(L)
print M
```

the matrix of evaluated monomials

$$
\begin{pmatrix}
\frac{-2187}{10} & \frac{-419904}{625} & \frac{2916}{25} & \frac{-81}{4} & \frac{-7776}{125} & \frac{54}{5} & \frac{20736}{625} & \frac{-144}{25} & 1 \\
\frac{-80}{3} & \frac{-1875}{16} & 25 & \frac{-16}{3} & \frac{-375}{16} & 5 & \frac{5625}{256} & \frac{-75}{16} & 1 \\
\frac{-2}{3} & \frac{-512}{81} & \frac{16}{9} & \frac{-1}{2} & \frac{-128}{27} & \frac{4}{3} & \frac{1024}{81} & \frac{-32}{9} & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{81}{16} & \frac{-9}{4} & 1 \\
\frac{2048}{3} & 48 & 64 & \frac{256}{3} & 6 & 8 & \frac{9}{16} & \frac{3}{4} & 1 \\
\frac{15625}{6} & \frac{40000}{81} & \frac{2500}{9} & \frac{625}{4} & \frac{800}{27} & \frac{50}{3} & \frac{256}{81} & \frac{16}{9} & 1 \\
\frac{34992}{5} & \frac{32805}{16} & 729 & \frac{1296}{5} & \frac{1215}{16} & 27 & \frac{2025}{256} & \frac{45}{16} & 1 \\
\frac{235298}{15} & \frac{3687936}{625} & \frac{38416}{25} & \frac{2401}{6} & \frac{18816}{125} & \frac{196}{5} & \frac{9216}{625} & \frac{96}{25} & 1
\end{pmatrix}
$$

the kernel of the matrix

The sympy script continues as

```
N = M.nullspace()
print sp.Matrix(N).transpose()
equ = 0
for k in range(len(polygon)):
    mon = polygon[k]
    equ = equ + N[0][k]*x**mon[0]*y**mon[1]
print equ
```

and prints

```
[1/81, -1/81, -5/81, -2/81, -4/81, -11/27, 16/81, 8/9, 1]
x**3*y**2/81 - x**2*y**3/81 - 5*x**2*y**2/81 - 2*x**2*y/81
- 4*x*y**2/81 - 11*x*y/27 + 16*y**2/81 + 8*y/9 + 1
```

the condition number

Normalizing *the* implicit equation to be monic:

$$x^3y^2 - x^2y^3 - 5x^2y^2 - 2x^2y - 4xy^2 - 33xy + 16y^2 + 72y + 81.$$

The `sympy` script continued

```
A = M[0:8, 1:9]
print A.det()
L = sp.Matrix.tolist(A)
import numpy as np
B = np.matrix([[float(x) for x in e] for e in L])
print np.linalg.norm(B)*np.linalg.norm(np.linalg.inv(B))
```

prints

```
3674636522691897/2000000
255962.285173
```

The numerical conditioning of this problem gets bad very quickly.

Tropical Curves and Amoebas

computing the Newton polygon

Tropical Implicitization Problem:
Given two rational functions $x = \phi_1(t)$ and $y = \phi_2(t)$,
compute the Newton polygon of the implicit equation $f(x, y) = 0$.

Rewrite

$$\Phi(t) = \left(\frac{t^3 + 4t^2 + 4t}{t^2 - 1}, \frac{t^3 - t^2 - t + 1}{t^2} \right).$$

in factored form:

$$\begin{array}{rcccccccc} x & = & \phi_1(t) & = & (t-1)^{-1} & t^1 & (t+1)^{-1} & (t+2)^2 \\ y & = & \phi_2(t) & = & (t-1)^2 & t^{-2} & (t+1)^1 & (t+2)^0. \end{array}$$

Recall from the amoebas: edges of the Newton polygon are where the algebraic curve meets the coordinate axis and/or infinity.

We see that x and y go to 0 or ∞ when t goes to a root.

collecting exponents

Considering the logarithms of the absolute values of

$$\begin{array}{rcllll} \phi_1(t) & = & (t-1)^{-1} & t^1 & (t+1)^{-1} & (t+2)^2 \\ \phi_2(t) & = & (t-1)^2 & t^{-2} & (t+1)^1 & (t+2)^0 \end{array}$$

gives

$$\begin{array}{rcllll} \log|\phi_1(t)| & = & -1 \ \log|t-1| & +1 \ \log|t| & -1 \ \log|t+1| & +2 \ \log|t+2| \\ \log|\phi_2(t)| & = & 2 \ \log|t-1| & -2 \ \log|t| & +1 \ \log|t+1| & +0 \ \log|t+2|. \end{array}$$

So we collect the exponents of Φ:

$$\left(\begin{array}{r} -1 \\ 2 \end{array} \right), \left(\begin{array}{r} 1 \\ -2 \end{array} \right), \left(\begin{array}{r} -1 \\ 1 \end{array} \right), \left(\begin{array}{r} 2 \\ 0 \end{array} \right).$$

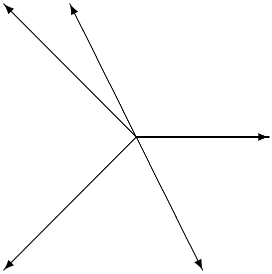
The exponents correspond to the rays in the tropical plot.

the balancing condition

The exponents need to add up to zero:

$$\begin{pmatrix} -1 \\ 2 \end{pmatrix} + \begin{pmatrix} 1 \\ -2 \end{pmatrix} + \begin{pmatrix} -1 \\ 1 \end{pmatrix} + \begin{pmatrix} 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

So we add $(-1,-1)^T$ as another ray. Below is a plot of the rays.



vector rotation over 90 degrees

The rays are perpendicular to the edges of the Newton polygon.

The given vectors

$$\left(\begin{array}{r} -1 \\ -1 \end{array}\right), \left(\begin{array}{r} -1 \\ 2 \end{array}\right), \left(\begin{array}{r} 1 \\ -2 \end{array}\right), \left(\begin{array}{r} -1 \\ 1 \end{array}\right), \left(\begin{array}{r} 2 \\ 0 \end{array}\right)$$

are rotated over 90 degrees clockwise:

$$\left(\begin{array}{r} -1 \\ 1 \end{array}\right), \left(\begin{array}{r} 2 \\ 1 \end{array}\right), \left(\begin{array}{r} 2 \\ -1 \end{array}\right), \left(\begin{array}{r} 1 \\ 1 \end{array}\right), \left(\begin{array}{r} 0 \\ -2 \end{array}\right).$$

Observe that also the rotated vectors sum up to zero.

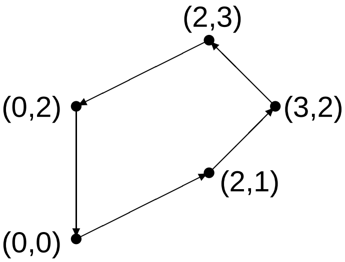
Because of the zero sum,
there is a polygon that has those vectors as edges.

sorting and concatenating vectors

We sort the vectors by increasing slope:

$$\begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \begin{pmatrix} -2 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ -2 \end{pmatrix}.$$

and concatenate them starting at the origin:



an algorithm to compute the Newton polygon

Input: $\Phi(t) : (x = \phi_1(t), y = \phi_2(t))$, ϕ_1 and ϕ_2 are rational polynomials.

Output: vertex points that span the Newton polygon of $f(x, y) = 0$,
where f is the irreducible polynomial of the image of Φ.

0. Apply the Euclidean algorithm on ϕ_1 and ϕ_2 to compute the rays.
   (*Be aware of multiplicities when applying a root finder.*)
1. Rotate the rays clockwise over 90 degrees.
2. Sort the rotated vectors by increasing slope.
3. Concatenate the sorted vectors starting at the origin.
4. The end points of the concatenated vectors are the vertex points of the Newton polygon of the implicit equation $f(x, y) = 0$.

applying the fundamental theorem of tropical geometry

Theorem

The tropical curve $V(f)$ defined by the unknown polynomial f coincides with the tropical curve determined by the rays computed from Φ.

This theorem is a direct result from the fundamental theorem of tropical geomety, proved in Chapter 3.

Corollary

The polygon P coincides with the Newton polygon of the defining irreducible polynomial f of the curve defined by the image of Φ.