COMPUTING ZERO-DIMENSIONAL TROPICALIZATIONS IN OSCAR

ARMAN MARTI-SHAHANDEH, YUE REN, VICTORIA SCHLEIS

Abstract. This document is solely for purpose of highlighting the outcomes of the summer project, with a formal preprint and code release expected in mid-2025. Here, we showcase the soon-to-be released OSCAR package `zero_dim_tropicalization.jl`, for which the active development can be found at https://github.com/armanmarti-shahandeh/Zerodimensional-Tropicalisation, which provides functionality for computing zero-dimensional varieties over local fields using tropical geometry.

1. Introduction

Tropical varieties are commonly described as a combinatorial shadow of their algebraic counterparts, and they play a central role in the area of tropical geometry. Their computation plays an important role both in research on tropical geometry as well as in applications of tropical geometry. For example showing that the tropicalization equals $\{0\}$ proves the finiteness of certain central configurations in celestial mechanics [HM06; HJ11], and computing tropicalizations give rise to better ways to solve certain polynomial systems [HHR24].

Fundamentally, given polynomial ideal $I \subseteq K[x_1, \dots, x_n]$ and a possibly trivial valuation $\mathrm{val}\colon K^* \to \mathbb{R}$, the task is to find a set of polyhedra covering $\mathrm{Trop}(I) \subseteq \mathbb{R}^n$. This is generally done via a traversal over $\mathrm{Trop}(I)$ [BJSST07; MR20], and most computations in that traversal can be boiled down computations of zero-dimensional tropical varieties [HR18], i.e., the case where I is zero-dimensional and $\mathrm{Trop}(I)$ consists of finitely many points.

In this paper, we report on a new implementation of the algorithm for computing zero-dimensional tropical varieties in OSCAR [OSCAR], with computation of zero-dimensional ideals and varieties lying at the heart of many algorithms in computer algebra. We use triangular decomposition and root approximation, as in [HR18], however our implementation has much finer control on the precision required for computing $\mathrm{Trop}(I)$. We further make generous use of the `algebraic_closure` functionality in OSCAR, efficiently automating the handling of iterated field extensions behind a layer of abstraction, making our code more readable, as well as more versatile in application.

Date: February 2025.

Funding. Y.R. and V.S. were funded by the UKRI FLF *Computational Tropical Geometry* MR/Y003888/1. Additionally, funding for A.M.S was provided for by the Hatfield Trust Undergraduate Research Scholarship. During parts of this project, V.S. was a member of the Institute for Advanced Study, funded by the Charles Simonyi Endowment.

2. Zero-dimensional tropical varieties and their generating sets

In the following, let K be an algebraically closed field with a non-trivial valuation $\mathrm{val}\colon K^* \to \mathbb{R}$ and write $z \in K$. We write $\mathfrak{K}$ for the residue field of K, and $\overline{z}$ for the residue of z in $\mathfrak{K}$. Further, we usually write $\tilde{z} = z + O(t^r)$ for an approximate element of K, and use $r \in \mathbb{R}$ for its precision. We use $p \in \mathbb{R}$ to denote the precision of polynomials.

2.1. **Newton polygons.** In this section, we will discuss initial forms and ideals, focused on defining triangular sets and their extended Newton polyhedra. For initial ideals, our main references are [CLO15; MS15], while we refer to [HR18; Jos21] for triangular sets and their Newton polyhedra.

Let $f = \sum_{u\in\mathbb{N}^n} c_u x^u \in K[x]$ be a polynomial, and let $w \in \mathbb{R}^n$. Its *initial form* is defined as

$$\mathrm{in}_w(f) = \sum_{\substack{w\cdot u+\mathrm{val}(c_u)=\\ \min(w\cdot u+\mathrm{val}(c_u)|c_u\neq 0)}} \overline{c_u t^{-\mathrm{val}(c_u)}} x^u \in \mathfrak{K}[x].$$

For an ideal $I \subseteq K[x]$, we define the *initial ideal* $\mathrm{in}_w(I) = \langle \mathrm{in}_w(f) | f \in I\rangle \subseteq \mathfrak{K}[x]$. Initial ideals are important for defining the structure of tropical varieties and used extensively through our use of Gröbner bases.

Definition 2.1 For a univariate polynomial $f = \sum_{u=0}^d c_u x^u \in K[x_i]$, the *Newton polygon* is the set

$$\Delta(f) = \mathrm{conv}(u, \mathrm{val}(c_u)|c_u \neq 0) + (\{0\} \times \mathbb{R}_{\geq 0}) \subseteq \mathbb{R}^2$$

For a weight $w \in \mathbb{R}^{k-1}$ and a multivariate polynomial $f = \sum_{i=1}^d f_i \cdot x_k^i \in K[x]$ where all $f_i \in K[x_1, \dots, x_{k-1}]$, we define its multivariate analogue, the *expected Newton polygon*

$$\Delta_w(f) = \mathrm{conv}(\{i, \min_{c_u\in f_i}(w\cdot u + \mathrm{val}(c_u)|c_u \neq 0) + (\{0\} \times \mathbb{R}_{\geq 0}) \subseteq \mathbb{R}^2\})$$

We say that a Newton polygon is *well defined* if the initial form $\mathrm{in}_w(f_i)$ is a monomial at all vertices. Further, we denote by $\Lambda(f)$ (resp. $\Lambda_w(f)$) the set of negative slopes.

Note that the Newton polygons we consider here differ from the Newton polytope commonly used in tropical geometry!

2.2. **Tropical Geometry.** In this section, we review the basics of tropical geometry. Our main references are [MS15] and [Jos21]. In this paper we usually consider zero dimensional ideals I and write $V(I) \subseteq (K^*)^n$ for their corresponding zero-dimensional algebraic variety.

Definition 2.2 The *tropical hypersurface* of a tropical polynomial $F = \bigoplus_{u \in \mathbb{N}^n} c_u \odot x^u \in \mathbb{T}[x_1, \dots, x_n]$ is

$$V(F) = \left\{ x \in \mathbb{R}^n/\mathbb{R}\mathbf{1}) : \min_{u \in \mathbb{N}^n} \left\{ c_u + \sum_{i=1}^n u_i \cdot x_i \right\} \text{is achieved at least twice} \right\}.$$

For an ideal $J \subseteq \mathbb{T}[x_1, \dots, x_n]$ over the tropical polynomial semiring, we define the *tropical variety* to be the set

$$V(J) = \bigcap_{F \in J} V(F) \subseteq \mathbb{R}^n/\mathbb{R}\mathbf{1}).$$

The *tropicalization* of a polynomial $f = \sum_{u \in \mathbb{N}^n} a_u x^u \in K[x_1, \dots, x_n]$ is the tropical polynomial

$$\operatorname{trop}(f) = \bigoplus_{u \in \mathbb{N}^n} \operatorname{val}(a_u) \odot x^u \in \mathbb{T}[x_1, \dots, x_n].$$

Over an algebraically closed base field K with a non-trivial valuation, the *tropicalization* $\operatorname{trop}(X)$ of a subvariety $X \subseteq (K^*)^n$, is defined by

$$\operatorname{trop}(X) = \overline{\{(\operatorname{val}(x_0), \dots, \operatorname{val}(x_n)) \in \mathbb{R}^n/\mathbb{R}\mathbf{1}) : (x_1, \dots, x_n) \in X\}}.$$

The above definitions indicate that for an ideal I over an algebraically closed base field with non-trivial valuation, there are two ways to construct a tropicalization: we can either first construct the locus of its polynomials and then take the valuation of points, or we can first tropicalize the polynomials in the ideal and then intersect their tropical hypersurfaces. The *fundamental theorem of tropical geometry*, [MS15, Theorem 3.2.3], assures us that these two definitions of tropicalization coincide, i.e. that $\operatorname{trop}(V(I)) = V(\operatorname{trop}(I))$. Further, it allows us an additional characterization using initial forms discussed in the previous section - $\mathbf{w} \in \operatorname{trop}(V(I))$ if and only if its corresponding initial form $\operatorname{in}_{\mathbf{w}}$ is not a monomial.

3. Computing tropicalizations of triangular sets

In this section, we will describe the algorithms used for computing the tropicalization of a zero-dimensional ideal given by a special generating set.

Definition 3.1 A *triangular set* $\{f_1, \dots, f_n\} \in K[x_1^{\pm}, \dots, x_n^{\pm}]$ is a set of Laurent polynomials satisfying $f_i \in K[x_1^{\pm}, \dots, x_i^{\pm}]$, i.e., the ith polynomial is written only using the first i variables.

For every ideal I a triangular generating set can be obtained by computing a lexicographical Gröbner basis.

3.1. **The tree of a triangular set.** The global state of the computation is modelled using a tree which starts as a single vertex and grows to depth n and width $|V(F)|$. In depth i, each node represents an (approximate) root of $f_i(z_1,\dots,z_{i-1},x_i) \in K\{\{t\}\}[x_i]$, where $z_1,\dots,z_{i-1}$ are the roots above it.

Definition 3.2 We consider solution sets $V(F) \subseteq K^n$ as multisets where each root is listed with its appropriate multiplicity.

For each $z \in V(F)$ we fix $k_z \in [n]$ and let $\pi_{k_z}\colon K^n \twoheadrightarrow K^{k_z}$ denote the projection onto the first k_z coordinates. The resulting set $V_{\mathbf{k}}(F) := \{\pi_{k_z}(z) \mid z \in V(F)\}$ is a *partial solution set* of F. It can be thought of a snapshot of the solving process of F via back-substitution.

For our purposes, it is helpful to represent $V_{\mathbf{k}}(F)$ by a *partial solution tree*, which is a finite, labelled, rooted tree $\Gamma_{\mathbf{k}}(F)$ such that

(1) the root vertex $v_0 \in V(\Gamma_{\mathbf{k}}(F))$ is labelled by the symbol x_0,
(2) each non-root vertex $v_i \in V(\Gamma_{\mathbf{k}}(F))$ is labelled by a variable x_i,
(3) edges $(v_{i-1},v_i) \in E(\Gamma_{\mathbf{k}}(F))$ connect consecutively labelled vertices,

together with a map $z\colon V(\Gamma_{\mathbf{k}}(F)) \to K, v \mapsto z_v$ such that we have a bijection

$$\begin{array}{ccc} \mathrm{Branches}(\Gamma_{\mathbf{k}}(F)) & \longleftrightarrow & V_{\mathbf{k}}(F) \\ \big((v_0,v_1),(v_1,v_2),\dots,(v_{l-1},v_l)\big) & \longmapsto & (z(v_1),\dots,z(v_l)). \end{array}$$

Here, a branch refers to a chain of edges $(v_0,v_1),(v_1,v_2),\dots,(v_{l-1},v_l) \in E(\Gamma_{\mathbf{k}})$ connecting the root v_0 to a leaf v_l, i.e., a vertex of degree 1.

Definition 3.3 Given a triangular set $F = \{f_1,\dots,f_n\} \subseteq K[x]$ and the corresponding uncertainty ring $K[u][x]$, a *partial solution tree* is a finite, labelled, rooted tree Γ such that

(1) the root vertex $v_0 \in V(\Gamma_{\mathbf{k}}(F))$ is labelled by the symbol x_0,
(2) each non-root vertex $v_i \in V(\Gamma_{\mathbf{k}}(F))$ is labelled by a variable x_i,
(3) edges $(v_{i-1},v_i) \in E(\Gamma_{\mathbf{k}}(F))$ connect consecutively labelled vertices,

together with two maps

$$z\colon V(\Gamma) \to K[u], v \mapsto z_v,$$
$$p\colon V(\Gamma) \to \mathbb{Q}_{\geq 0}, v \mapsto p_v.$$

Here, the job of z is to map a vertex v_i to an approximate root of $f(z(v_1),\dots,z(v_{i-1}))$ in $K[u]$ where $(v_1,\dots,v_i)$ is a branch of Γ. The job of p is two-fold: If $p(v_i) > 0$, then $p(v_i)$ is the relative precision of z_1 used to compute z_i. If $p(v_i) = 0$, then z_i was computed via the slopes of the Newton polygon and is thus of the form $z_i = u_i \cdot t^{w_i}$ for some $w_i \in \mathbb{Q}$.

We can initialize the partial solution tree of a triangular set F using the algorithm `init_tree` 3.4 below. This algorithm constructs the first level of the branched tree, corresponding to the roots of the univariate polynomial f_1 in the triangular set.

Algorithm 3.4 (`init_tree`)
Input: $F = \{f_1, \dots, f_n\} \subseteq K[x^{\pm}]$ a triangular set
Output: $\Gamma_{\mathbf{1}}$ a partial solution tree
1: Let $\Gamma_{\mathbf{1}}$ be a tree with root vertex v_0 and one leaf v_w per $w \in \mathrm{Trop}(f_1)$.
2: Let $z\colon V(\Gamma_{\mathbf{1}}) \to K, v_w \mapsto O(t^w)$.
3: **return** $(\Gamma_{\mathbf{1}}, z)$

Definition 3.5 Let $f \in K[x_1, \dots, x_k]$ and $\tilde{z}_1, \dots, \tilde{z}_{k-1} \in K$ be approximate points, i.e., $\tilde{z}_j = z_j + O(t^{r_j})$ for some $z_j \in K$ and some $r_j \in \mathbb{R}$ with $\mathrm{val}(z_j) < r_j$.

Let $f' := f(z_1 + t^{r_1}x_1 \dots, z_{k-1} + t^{r_{k-1}}x_{k-1}, x_k)$, say $f' = \sum_{i=0}^{d_k} f_i' x_k^i$ for some $f_i \in K[x_1, \dots, x_{k-1}]$, and let $\mathbf{0} := (0, \dots, 0) \in \mathbb{R}^{k-1}$. Consider the *expected Newton polygon*

$$\Delta' := \mathrm{conv}\left(\left\{(i, \mathrm{trop}(f_i')(\mathbf{0})) \mid i = 0, \dots, d_k\right\}\right) + \mathbb{R}_{\geq 0} \cdot (0, 1) \subseteq \mathbb{R}^2.$$

We say *the Newton polygon of* $f(\tilde{z}_1, \dots, \tilde{z}_{i-1})$ *is well-defined*, or $\Delta(f(\tilde{z}_1, \dots, \tilde{z}_{i-1}))$ *is well-defined* if for all vertices $(i, \mathrm{trop}(f_i')(\mathbf{0})) \in V(\Delta')$ we have $\mathrm{in}_{\mathbf{0}}(f_i') \in K$ is constant.

Now, we can construct the tropicalization of a triangular set using the algorithm `trop_triangular` below. This algorithm takes a triangular set and a specified maximal precision and constructs the associated solution tree, from which the tropicalization can be read off. It does so by taking the initial tree constructed by `init_tree` and recursively constructing new leaves of the partial solution tree.

If the extended Newton polygon $\Delta(f_i)$ at a step is not well-defined, the algorithm will try to remedy this by increasing the precision of computations and re-computing the Newton polygon. This process will be repeated until either the branch gets completed, a solution branch is found to be infeasible, or the computation would exceed the maximum precision.

Algorithm 3.6 (`trop_triangular`)
Input: $F = \{f_1, \dots, f_n\} \subseteq K[x^{\pm}]$ a triangular set, $w_{\max} \in \mathbb{R}$ a maximal precision.
Output: $\mathrm{Trop}(\langle F \rangle) \subseteq \mathbb{R}^n$, the tropicalization of F.
1: Initialise $(\Gamma, z) := $ `InitTree`(F).
2: **while** (Γ, z) incomplete **do**
3: Pick a branch $(v_0, v_1, \dots, v_{i-1}, v_i)$.
4: **while not** `is_newton_polygon_well_defined`$(f_{i+1}(z(v_1), \dots, z(v_i)))$ **do**
5: **if** current precision of $z(v_1)$ is the precision used for $z(v_i)$ **then**
6: Increase precision of $z(v_1)$. Return error if precision exceeds $w_{\max}$.

    Update $z(v_2), \dots, z(v_i)$.
  for $w \in \mathrm{Trop}(f_{i+1}(z(v_1), \dots, z(v_i)))$ **do**
    Attach a new vertex $v_{i+1,w}$ to v_i.
    Set $z(v_{i+1,w}) \coloneqq \texttt{local\_field\_expansion}(f_{i+1}(z(v_1), \dots, z(v_i)), w)$.
Read off $\mathrm{Trop}(\langle F \rangle)$ from Γ:

$$\mathrm{Trop}(\langle F \rangle) \coloneqq \Big\{ \mathrm{val}((z(v_1), \dots, z(v_n))) \;\Big|\; (v_1, \dots, v_n) \text{ branch of } \Gamma \Big\}.$$

return $\mathrm{Trop}(\langle F \rangle)$

3.2. **Roots of univariate polynomials.** The main algorithm of the previous section, Algorithm 3.6, repeatedly requires the computation of roots of univariate polynomials over a local field. The following algorithm constructs all roots of a polynomial f over a local field with valuation w by explicitly computing all possible coefficients in front of t^w and recursively calling itself with valuations $w' > w$.

Algorithm 3.7 (`local_field_expansion`)

Input: (f, w, b), where
  (1) $f \in K[u][x_i]$ is a univariate polynomial in x_i with coefficients in the uncertainty ring $K[u]$,
  (2) $w \in \mathbb{R}$ is a desired tropical point,
  (3) $b \in \mathbb{R}$ is a precision computation stopgap,

Output: $\{\tilde{z} \in V(f) \mid \mathrm{val}(\tilde{z}) = w\}$ up to precision b.

Set $h \coloneqq \mathrm{in}_0(\tilde{f}(t^w \cdot x))$.
if $w \geq b$ **or** $h \notin \mathfrak{K}[x_i] \subseteq \mathfrak{K}[u][x_i]$ **then**
  return $\{u_i \cdot t^w\}$
Set $f_{ct^w} \coloneqq f|_{x_i = x_i + c \cdot t^w} \in K[u][x_i]$ for $c \in V(h) \subseteq \mathfrak{K}$, $c \neq 0$.
return

$$\Bigg(\bigcup_{\substack{c \in V(h) \\ c \neq 0}} \; \bigcup_{\substack{w' \in \mathrm{Trop}(\tilde{f}_{ct^w}) \\ w' > w}} \Big\{ c \cdot t^w + z \;\Big|\; z \in \texttt{local\_field\_expansion}(f_{ct^w}, w', b) \Big\} \Bigg)$$
$$\cup \Big\{ c \cdot t^w \;\Big|\; c \in V(h) \text{ with } c \neq 0 \text{ and } f_{ct^w}(0) = 0 \Big\}$$

Example 3.8 We now demonstrate Algorithm 3.7 for the polynomial

$$f(x) = x^2 - x(2 + t + 2t^2 + 2t^3 + \mathcal{O}(t^4)) + (1 + t + 2t^2 + 3t^3 + \mathcal{O}(t^4)),$$

the tropical point $w = 0$, and maximal precision $b = 3$. We further set $w_{prev} = 0$.

Since the Newton polygon is uniquely defined and $w < b$, in the first iteration, we do the following:

- Set $h \coloneqq \mathrm{in}_0(f(x \cdot t^0)) = (x - 1)^2$.
- Then $V(h) = \{1\}$, and $1 \cdot t^0$ is the first term of our root(s).
- Set $f_{1 \cdot t^0} \coloneqq f(1 \cdot t^0 + x)$.

- ◦ Calculate $\mathrm{Trop}(f_{1\cdot t^0}) = \{1, 2\}$ and note that both $w' = 2 > 0$, $w' = 1 > 0$.

From here, the recursion branches – we have to carry out the same computation for both $w' = 1$ and $w' = 2$.

For $w' = 2$, we set $h := \mathrm{in}_0(f_{1\cdot t^0}(x\cdot t^2)) = 1-x$. Then, $V(h) = \{1\}$, and $1\cdot t^2$ is the next term of our root(s). Thus, we set $f_{1+1\cdot t^2} := f_{1\cdot t^2}(f_{1\cdot t^0}) := f(1+1\cdot t^2+x)$. When calculating $\mathrm{Trop}(f_{1+1\cdot t^2}) = \{1, 3\}$, only $w' = 3 > 2$, so this is the only recursion branch we can follow. Notice that $3 = b$. This implies that our root computation was already sufficiently precise, and we `return` $1 + t^2$.

Now, we return to the top of the recursion branch. For $w' = 1$, we set $h := \mathrm{in}_0(f_{1\cdot t^0}(x \cdot t)) = x^2 - x$. Now, $V(h) = \{0, 1\}$, and, as 0 is trivial, the next term of our root is $1 \cdot t$. Hence, we set $f_{1+1\cdot t^1} := f_{1\cdot t^1}(f_{1\cdot t^0}) := f(1 + 1 \cdot t + x)$. Again, computing the tropicalization $\mathrm{Trop}(f_{1+1\cdot t^1}) = \{1, 2\}$, we note that only $w' = 2 > 1$, thus, this is our new recursion branch. Applying the algorithm to this branch, we obtain that $1 \cdot t^2$ is the next term of our root. Further, $\mathrm{Trop}(f_{1+1\cdot t^1+1\cdot t^2}) = \{1, 3\}$, so the algorithm terminates and we `return` $1 + t + t^2$.

4. Further Project Objectives

There are multiple aspects of the project which are continually being built out, including but not limited to:

- Contained run-times and stress-testing of the code through various "mother examples", which illustrate the best and worst case scenarios for Algorithm 3.6.
- Definition and explanation of many of the algebraic sub-tools, as well as combinatorial book-keeping structures, developed and utilised throughout the implementation.
- Timings and efficacy measures when using the algorithm on external/applied mathematical problems.
- Further functionality for the "uncertainty ring", over which the local field polynomials live, which will allow it to detect and flag imprecision in the base ring itself.
- Porting of basis-triangulation tools from `Singular`, to enable calculation on generalised ideals.

References

[BJSST07] T. Bogart, A. N. Jensen, D. Speyer, B. Sturmfels, and R. R. Thomas. "Computing Tropical Varieties". In: *Journal of Symbolic Computation* 42.1-2 (2007), pp. 54–73. doi: 10.1016/j.jsc.2006.02.004.

[CLO15] D. A. Cox, J. Little, and D. O'Shea. *Ideals, varieties, and algorithms.* Fourth. Undergraduate Texts in Mathematics. An introduction to computational algebraic geometry and commutative algebra. Springer, Cham, 2015.

[HJ11] M. Hampton and A. Jensen. "Finiteness of Spatial Central Configurations in the Five-Body Problem". In: *Celestial Mechanics and Dynamical Astronomy* 109.4 (Apr. 2011), pp. 321–332. DOI: 10.1007/s10569-010-9328-9.

[HM06] M. Hampton and R. Moeckel. "Finiteness of Relative Equilibria of the Four-Body Problem". In: *Inventiones mathematicae* 163.2 (Feb. 2006), pp. 289–312. DOI: 10.1007/s00222-005-0461-0.

[HHR24] P. A. Helminck, O. Henriksson, and Y. Ren. *A Tropical Method for Solving Parametrized Polynomial Systems.* Sept. 2024. DOI: 10.48550/arXiv.2409.13288. eprint: 2409.13288 (math).

[HR18] T. Hofmann and Y. Ren. "Computing Tropical Points and Tropical Links". In: *Discrete & Computational Geometry* 60.3 (Oct. 2018), pp. 627–645. DOI: 10.1007/s00454-018-0023-z.

[Jos21] M. Joswig. *Essentials of tropical combinatorics.* Vol. 219. Graduate Studies in Mathematics. American Mathematical Society, Providence, RI, 2021.

[MS15] D. Maclagan and B. Sturmfels. *Introduction to tropical geometry.* Vol. 161. Graduate Studies in Mathematics. American Mathematical Society, Providence, RI, 2015.

[MR20] T. Markwig and Y. Ren. "Computing Tropical Varieties over Fields with Valuation". In: *Foundations of Computational Mathematics* 20.4 (2020), pp. 783–800. DOI: 10.1007/s10208-019-09430-2.

[OSCAR] *OSCAR – Open Source Computer Algebra Research system, Version 1.0.0.* The OSCAR Team, 2024. URL: https://www.oscar-system.org.