Algorithms in Tropical Geometry

by

Leon Zhang

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Mathematics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Bernd Sturmfels, Chair
Professor David Eisenbud
Professor John Huelsenbeck

Spring 2021

Algorithms in Tropical Geometry

Abstract

Algorithms in Tropical Geometry

by

Leon Zhang

Doctor of Philosophy in Mathematics

University of California, Berkeley

Professor Bernd Sturmfels, Chair

Tropical geometry is an emerging field with strong connections in a wide array of areas both inside and outside mathematics. Efficient algorithms in tropical geometry take on a particular importance for developing these applications to other fields and for providing access to non-experts. The purpose of this dissertation is to provide novel and effective algorithms for the computation of various tropical objects of interest in pure mathematics and in phylogenetic data analysis.

We begin in the first chapter by describing some basic notions in tropical geometry which arise repeatedly throughout this thesis. We introduce tropical hypersurfaces, varieties, and prevarieties. We then discuss two classes of objects which are tropically convex, tropical linear spaces and tropical polytopes, and the ways in which they serve as tropical analogues to their classical equivalents. We also detail the connection between tropical geometry and the space of phylogenetic trees.

In the second chapter we discuss the computation of implicit tropicalizations of a very affine curve. We reduce the problem to the calculation of a basis for the group of units in the coordinate ring of the variety. We describe practical algorithms for computing such a basis for rational normal curves and elliptic curves. Our approach is rooted in divisor theory, based on interpolation in the case of rational curves and on methods from algebraic number theory in the case of elliptic curves.

In the third chapter we study the problem of computing the tropicalization of zero-dimensional varieties, which is a fundamental component of some algorithms for computing general tropical varieties. Our approach is via projections of the variety onto well-chosen lines, which are used to reconstruct the original variety. Our main algebraic tools for doing this are fast monomial transforms of triangular sets. Given a Gröbner basis, we show that our algorithms require only a polynomial number of arithmetic operations, and for ideals in shape position we demonstrate that the algorithm performs well in practice against alternative approaches.

In the fourth chapter we describe the first time-bounded complexity algorithms for realizing the min-convex hull of a finite collection of points in the affine building of SL_d as a tropical polytope lying inside a tropical linear space. These min-convex hulls describe the relations among a finite collection of invertible matrices over a field with valuation. As a consequence, we obtain a bound on the dimension of the tropical projective space needed to realize the min-convex hull as a tropical polytope.

In the fifth chapter we describe two tropical analogues of principal component analysis for the dimensionality reduction of ultrametric tree datasets. In one approach, we study the Stiefel tropical linear space of fixed dimension closest to the data points in the tropical projective torus; in the other, we consider the tropical polytope with a fixed number of vertices closest to the data points. We relate tropical best-fit hyperplanes to the tropical volume and prove that the polytropal decomposition of a tropical polytope spanned by ultrametrics refines the decomposition of the polytope into different tree topologies. We also give heuristic algorithms for both approaches and apply them to phylogenetics, testing the methods on simulated phylogenetic data and on an empirical dataset of Apicomplexa genomes.

To my father

Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I would like to thank my advisor, Bernd Sturmfels, for his expansive enthusiasm and his steady guidance throughout my doctoral studies. Thank you to all of my collaborators: Madeline Brandt, Marie Charlotte-Brandenburg, Justin Chen, Sophia Elia, Christopher Eur, Miriam Farber, Paul Görlach, Charles Johnson, Guido Montúfar, Robert Page, Yue Ren, Melissa Sherman-Bennett, Sameera Vemulapalli, Ruriko Yoshida, and Xu Zhang. I learned so much from working with you. Thank you to Anna, Bo, Maddie, Madeline, Mahsa, Paul, and Tim, my academic siblings, for your guidance and companionship throughout graduate school. Thank you David Eisenbud and John Huelsenbeck for taking the time to be on my dissertation committee. Thank you to Zvezdelina Stankova and Kelli Talaska for helping me to grow as an instructor. Thank you to Vicky Lee and Isabel Seneca for your help through every administrative hurdle.

I would like to thank Alborz, Eric, Jack, James, Katherine, Ryan, and Yajit, with whom I developed my interest in science and mathematics in high school. Thank you to Alex, Daniel, Fermi, and Yajit for so many late nights working on problem sets in college, and for so many hours and conversations since. Thank you to Albert, Alborz, Amruth, Daniel, Fermi, Harini, Katherine, Nick, Nitya, Ryan, and Sophie. You made the Bay Area my home for the last six years.

Thank you to Melissa for sharing your life and keeping me anchored. Thank you to Lily: you have been a role model since the very beginning. Thank you to Luke for all the fun and all the joy. Thank you to Jian Wu: your love lives on in me. Finally, an enormous thank you to Zhenyu Zhang and Zhengyuan Ju, for your encouragement, support, and the opportunities you worked so hard to give me.

Chapter 1

Introduction

Tropical geometry is a piecewise-linear analogue to ordinary mathematics in which the operations of addition and multiplication are replaced with minimum and addition, respectively. Problems throughout mathematics and science involve the extraction of an extremal value from a set of real numbers. As a consequence, connections to tropical geometry arise both inside and outside pure mathematics, in areas as diverse as algebraic topology [16] and optimization [2] within mathematics as well as auction theory [3, 86] and celestial mechanics [42, 43] outside it. Efficient algorithms take on a particular importance for developing these applications to other fields and providing access to non-experts. The purpose of this dissertation is to provide novel and effective algorithms for the computation of various tropical objects of interest in pure mathematics and phylogenetics.

In this chapter, we describe the fundamentals of tropical mathematics on which we will rely throughout this thesis, largely following the treatment in [52] and [63]. In Section 1.1, we introduce the tropical semiring and review tropical hypersurfaces, varieties, and prevarieties. In Section 1.2 we dive deeper into a particular class of tropical prevarieties called tropical linear spaces. In Section 1.3 we recall the basics of the tropical analogue of convexity. In Section 1.4 we explore an application of tropical geometry to phylogenetics, via the space of ultrametric trees. Finally, in Section 1.5 we summarize the main novel results of this dissertation.

1.1 Tropical varieties

In this dissertation we generally work over the tropical semiring $\mathbb{T} = (\mathbb{R} \cup \{\infty\}, \oplus, \odot)$ with the min-plus convention. In this semiring, the basic arithmetic operations of addition and multiplication are redefined as follows:

$$a \oplus b := \min(a, b), \quad a \odot b := a + b \text{ for any } a, b \in \mathbb{T}.$$

So, for instance, $3 \oplus -5 = -5$ and $3 \odot -5 = -2$. The additive identity in $\mathbb{T}$ is ∞ and the multiplicative identity is 0.

Remark 1.1. *There is also an analogous* max-plus *tropical semiring* $(\mathbb{R} \cup \{-\infty\}, \oplus, \odot)$*, wherein addition is defined as* $a \oplus b := \max(a, b)$*. The min-plus and max-plus semirings are isomorphic and everything that follows in this chapter can be adapted for the max-plus tropical semiring. The max-plus semiring arises in many applications; we will accordingly switch to the max-plus convention in discussing a tropical analogue of principal component analysis in Chapter 5.*

Throughout this dissertation, we let K be a field with valuation $\nu\colon K^* \to \mathbb{R}$. For instance, we might have $K = \mathbb{Q}$ be the rational numbers with the 3-adic valuation, so $\nu(x)$ picks out the exponent of 3 in the irreducible form of x, as in $\nu\left(\frac{73}{27}\right) = -3$. Another example is the field $K = \mathbb{C}((t))$ of formal Laurent series, with the valuation ν picking out the exponent of the term of lowest order, as in $\nu(t^{-1} + (3+2i)t^2 + 7t^5) = -1$.

Definition 1.2. *Fix a multivariate Laurent polynomial ring* $K[\mathbf{x}^{\pm}] := K[x_1^{\pm}, \ldots, x_n^{\pm}]$*. Given a vector* $\mathbf{v} = (v_1, \ldots, v_n) \in \mathbb{Z}^n$*, we write* $\mathbf{x}^{\mathbf{v}} = x_1^{v_1} \cdot x_2^{v_2} \cdot \cdots \cdot x_n^{v_n}$*. Then the* tropicalization $\operatorname{trop}(f)$ *of a Laurent polynomial* $f = \sum_{\mathbf{v} \in \mathbb{Z}^n} a_{\mathbf{v}} \mathbf{x}^{\mathbf{v}} \in K[\mathbf{x}^{\pm}]$ *is obtained by replacing each* $a_{\mathbf{v}}$ *with its valuation* $\nu(a_{\mathbf{v}})$ *and sums and products with tropical sums and products respectively. Explicitly,*

$$\operatorname{trop}(f) = \bigoplus_{\mathbf{v} \in \mathbb{Z}^n} \nu(a_{\mathbf{v}}) \odot \mathbf{x}^{\odot \mathbf{v}} = \min_{\mathbf{v} \in \mathbb{Z}^n} (\nu(a_{\mathbf{v}}) + \mathbf{v} \cdot \mathbf{x}).$$

So, for example, if $K = \mathbb{Q}$ with the 3-adic valuation and $f = 6x^2 + 4xy + 27y^2 \in K[x^{\pm}, y^{\pm}]$, then $\operatorname{trop}(f) = \min(1 + 2x, x + y, 3 + 2y)$. Tropicalizations of polynomials allow us to define tropicalizations of hypersurfaces.

Definition 1.3. *Let* $f = \sum a_{\mathbf{v}} \mathbf{x}^{\mathbf{v}} \in K[\mathbf{x}^{\pm}]$ *be a Laurent polynomial over* K*. The tropicalization* $\operatorname{trop}(V(f))$ *is defined as*

$$\operatorname{trop}(V(f)) = \{\mathbf{x} \in \mathbb{R}^n \mid \min(\nu(a_{\mathbf{v}}) + \mathbf{v} \cdot \mathbf{x}) \text{ is attained at least twice}\}.$$

When f is homogeneous, the tropical hypersurface $\operatorname{trop}(V(f))$ contains a 1-dimensional lineality subspace spanned by the all-ones vector $\mathbf{1} := (1, 1, \ldots, 1)$. We often consider such tropical hypersurfaces and other tropical objects as lying inside the *tropical projective torus* $\mathbb{R}^n / \mathbb{R}\mathbf{1}$. When illustrating this space, as in Figure 1.4, we always choose the affine chart in which the first coordinate is 0.

Example 1.4. *Let* $K = \mathbb{C}((t))$ *be the field of formal Laurent series and* f *the polynomial* $f = 3t^{-1}x + 5y + 7t^{-2}z \in K[x^{\pm}, y^{\pm}, z^{\pm}]$*. Then* $\operatorname{trop}(f) = \min(x - 1, y, z - 2)$*, and* $\operatorname{trop}(V(f)) \subseteq \mathbb{R}^3 / \mathbb{R}\mathbf{1}$ *comprises the three infinite rays starting at the point* $(0, -1, 1)$ *illustrated in Figure 1.1. We can verify that the point* $(0, 1, 1)$ *lies in* $\operatorname{trop}(V(f))$*, as the minimum in* $\min(0, 1 + 1, 1 - 1)$ *is attained at least twice.*

In the univariate case, the tropical hypersurface of a Laurent polynomial $f = \sum_{i \in \mathbb{Z}} a_i x^i \in K[x^{\pm}]$ simply consists of the negated slopes of the Newton polygon of f [71, Proposition II.6.3], where the Newton polygon of f is the lower convex hull of the points $(i, \nu(a_i))$.

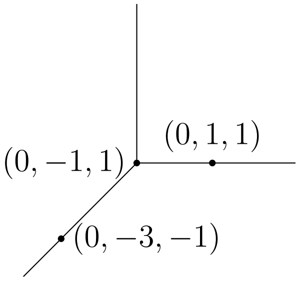


Figure 1.1: The tropical hypersurface trop($V(f)$) in Example 1.4.

Example 1.5. *Consider $K = \mathbb{Q}$ equipped with the 2-adic valuation and $f = 2x^4 + 3x^3 + 24x^2 + x + 12$. The Newton polygon of f is displayed in Figure 1.2; taking the negated slopes tells us that* $\operatorname{trop}(V(f)) = \{2, 0, -1\}$.

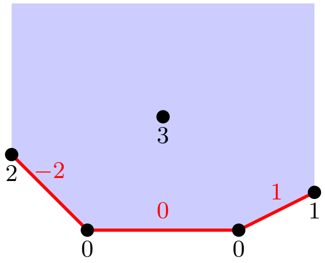


Figure 1.2: The Newton polygon of f in Example 1.5.

We can also define tropicalizations of general algebraic varieties. The following definition in terms of tropical hypersurfaces suffices for our purposes.

Definition 1.6. *Let $I \subseteq K[\mathbf{x}^{\pm}]$ be a Laurent polynomial ideal. The* tropical variety $\operatorname{trop}(I) \subseteq \mathbb{R}^n$ *is given by*

$$\operatorname{trop}(I) = \bigcap_{f \in I} \operatorname{trop}(V(f)),$$

the intersection of the tropical hypersurfaces corresponding to all polynomials in I.

Again, if I is homogeneous, we often consider the tropicalization $\operatorname{trop}(I)$ lying inside the tropcial projective torus $\mathbb{R}^n / \mathbb{R}\,\mathbf{1}$.

Remark 1.7. *The remarkable Fundamental Theorem of Tropical Geometry states that tropical varieties can also be defined in terms of coordinate-wise valuations of points in the original variety, or in terms of weight vectors whose corresponding initial ideal is monomial-free [63,* *Theorem 3.2.3]. We will not require the precise definitions in this dissertation so do not state the theorem in full.*

Theorem 1.8 ([63, Theorem 3.3.5])**.** *Let X be an irreducible d-dimensional subvariety of the torus T^n. Then* $\operatorname{trop}(X)$ *is the support of a balanced weighted polyhedral complex pure of dimension d.*

Note that while tropical hypersurfaces and varieties can be defined with multiplicities, we are largely only interested in them set-theoretically. Also, in general, if $f_1, \ldots, f_k$ generate an ideal $I \subseteq K[\mathbf{x}^{\pm}]$, it need not be the case that the intersection $\bigcap \operatorname{trop}(V(f_i))$ equals $\operatorname{trop}(I)$, or indeed any tropical variety at all. We call a finite intersection of tropical hypersurfaces a *tropical prevariety.*

1.2 Tropical linear spaces

Tropical linear spaces are a well-studied combinatorial class of tropical prevarieties. As the name suggests, tropical linear spaces provide a tropical twist on classical linear spaces: they arise via tropical analogues of matroids and Plücker relations, and in fact tropicalizations of ordinary linear spaces form an important subclass of tropical linear spaces. We now give a brief computational introduction to tropical linear spaces, adapting the treatment in [52, Chapter 10]. In what follows, let $[n]$ denote the set of integers $\{1, 2, \ldots, n\}$ where n is a positive integer.

Definition 1.9. *Let $p : \binom{[n]}{d} \to \mathbb{T}$ be a map from the set of all d-sized subsets of $[n]$ to $\mathbb{T}$. Take subsets σ and τ of $[n]$ of size $d-1$ and $d+1$ respectively, and suppose that the minimum*

$$\min_{i \in \tau} p(\sigma \cup \{i\}) + p(\tau - \{i\})$$

is attained at least twice, where by convention we say that $p(\gamma) = \infty$ if $|\gamma| < d$. If this property holds for all choices of σ and τ, we call p a valuated matroid *[24] or* tropical Plücker vector*.*

The definition above is simply the tropicalization of the corresponding quadratic Plücker relation for ordinary linear spaces [52, Equation 10.9]. As Plücker coordinates correspond to linear spaces, tropical Plücker coordinates give rise to tropical linear spaces.

Definition 1.10. *Let p be a tropical Plücker vector. The* tropical linear space L_p *consists of all points $x \in \mathbb{R}^n / \mathbb{R}\mathbf{1}$ such that, for any $(d+1)$-sized subset τ of $[n]$, the minimum of the numbers*

$$\min_{i \in \tau} p(\tau - \{i\}) + x_i$$

is attained at least twice.

Again, the tropical polynomials appearing in Definition 1.10 are tropicalizations of the circuit equations which cut out the ordinary linear space corresponding to a Plücker vector [63, Equation 4.3.6]. Tropical linear spaces are thus natural tropical-geometric analogues to ordinary linear spaces. Indeed, though they need not be tropical varieties, they are similarly well-behaved.

Theorem 1.11 ([63, Theorem 4.4.5]). *Let L_p be the tropical linear space corresponding to a tropical Plücker vector p. Then L_p is a balanced contractible polyhedral complex pure of dimension $d - 1$.*

Example 1.12. *Let $p : [4]^3 \to \mathbb{T}$ be defined as follows: $p(\{i, j, k\}) = \min(i, j, k)$ if $i \neq j \neq k$ and $p(\{i, j, k\}) = \infty$ otherwise. Then p is a tropical Plücker vector, and L_p is a 2-dimensional tropical plane as illustrated in Figure 1.3, consisting of six two-dimensional orthants all meeting at the corner $(0, 1, 1, 1)$.*

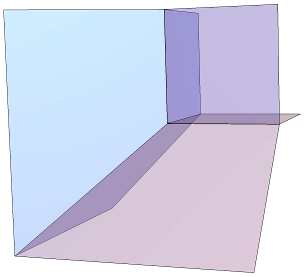


Figure 1.3: The tropical plane in Example 1.12.

One particularly well-behaved class of tropical linear spaces arises in analogy to the Stiefel map for ordinary linear spaces.

Definition 1.13. *Let A be an $m \times m$ matrix with entries in $\mathbb{T}$. We can define its* tropical determinant $\operatorname{tdet}(A)$ *in analogy with the classical operation as*

$$\operatorname{tdet}(A) = \bigoplus_{\sigma \in S_m} \left(\bigodot_{i=1}^{m} A_{i,\sigma(i)} \right).$$

If the tropical determinant of A is attained by at least two distinct permutations in S_m, the symmetric group on m elements, we say that A is tropically singular.

Definition 1.14. *Let A be a $d \times n$ matrix with entries in $\mathbb{T}$ such that $d \leq n$. Given a d-sized subset $\omega \subseteq [n]$, we write A_ω for the $d \times d$ matrix whose columns are the columns of A indexed by elements of ω. Then the map*

$$p : [n]^d \to \mathbb{R} \cup \{\infty\}$$

$$\omega \mapsto \operatorname{tdet}(A_\omega)$$

is a tropical Plücker vector. The corresponding tropical linear space is called the Stiefel tropical linear space *given by A.*

Example 1.15. *Let*

$$A = \begin{pmatrix} 0 & -3 & -1 \\ 1 & 0 & 3 \end{pmatrix}$$

and let p be its associated tropical Plücker vector. For ease of notation we write $p_{ij} = p(\{i,j\})$. Then

$$p_{12} = \operatorname{tdet}\begin{pmatrix} 0 & -3 \\ 1 & 0 \end{pmatrix} = -2,$$

$$p_{13} = \operatorname{tdet}\begin{pmatrix} 0 & -1 \\ 1 & 3 \end{pmatrix} = 0,$$

$$p_{23} = \operatorname{tdet}\begin{pmatrix} -3 & -1 \\ 0 & 3 \end{pmatrix} = -1.$$

The Stiefel tropical linear space corresponding to A is a tropical line in $\mathbb{R}^3 / \mathbb{R}\mathbf{1}$. It is the same hypersurface as in Example 1.4, pictured in Figure 1.1.

Throughout this dissertation, we will repeatedly want to project onto a tropical linear space. In order to describe a projection, we first need a notion of distance. In tropical geometry, this is given by the tropical metric.

Definition 1.16. *The tropical distance metric d_{tr} in $\mathbb{R}^n / \mathbb{R}\mathbf{1}$ is given by*

$$d_{tr}(v, w) := \max\{|v_i - v_j + w_j - w_i| : 1 \leq i < j \leq n\}, \tag{1.2.1}$$

which in essence compares the largest and smallest entries of the vector $v - w$.

Equipped with the tropical metric, we can now describe a projection operation onto a tropical linear space via the *Red and Blue Rules*. From [52, Proposition 10.76] we have:

Theorem 1.17 (The Blue Rule)**.** *Let $p : [n]^d \to \mathbb{T}$ be a tropical Plücker vector and $L_p \subseteq \mathbb{R}^n / \mathbb{R}\mathbf{1}$ the corresponding tropical linear space. Fix $u \in \mathbb{R}^n / \mathbb{R}\mathbf{1}$, and define the point $w \in \mathbb{R}^n / \mathbb{R}\mathbf{1}$ whose ith coordinate is*

$$w_i = \min_{\sigma} \max_{j \notin \sigma} (p(\sigma \cup \{i\}) - p(\sigma \cup \{j\}) + u_j),$$

where σ runs over all $(d-1)$-subsets of $[n]$ that do not contain i. Then $w \in L_p$, and any other $x \in L_p$ satisfies $d_{tr}(u, x) \geq d_{tr}(u, w)$. In other words, w attains the minimum tropical distance of any point in L_p to u.

Theorem 1.18 (The Red Rule)**.** *Let $p : [n]^d \to \mathbb{T}$ be a tropical Plücker vector and $L_p \subseteq \mathbb{R}^n / \mathbb{R}\mathbf{1}$ the corresponding tropical linear space. Fix $u \in \mathbb{R}^n / \mathbb{R}\mathbf{1}$ and let v be the all-zeros vector. For every $(d+1)$-sized subset τ of $[n]$, compute $\min p(\tau - \tau_i) + u_{\tau_i}$. Let γ_{τ,τ_i} be the nonpositive difference between this minimum and the second-minimum, and set $v_{\tau_i} = \max(v_{\tau_i}, \gamma_{\tau,\tau_i})$.*

Then v gives the difference between u and a closest point of L_p. In particular, if w is the point in L_p returned by the Blue Rule, we have

$$w = u + v.$$

We write π_{L_p} as the projection map which takes a point $u \in \mathbb{R}^n / \mathbb{R}\mathbf{1}$ and returns the nearest point $w \in L_p$ given by the Blue Rule. Depending on the size of d, one might prefer to use either the Blue Rule or the Red Rule to compute $\pi_{L_p}(u)$. We note that this closest point $\pi_{L_p}(u)$ computed by the Red and Blue Rules may not be unique; there may be other points in L_p which are of the same tropical distance from u.

Example 1.19. *Let A be the matrix of Example 1.15, with p and L_p its associated tropical Plücker vector and Stiefel tropical linear space. Let u be the point $(0, -2, 3) \in \mathbb{R}^3 / \mathbb{R}\mathbf{1}$. The*

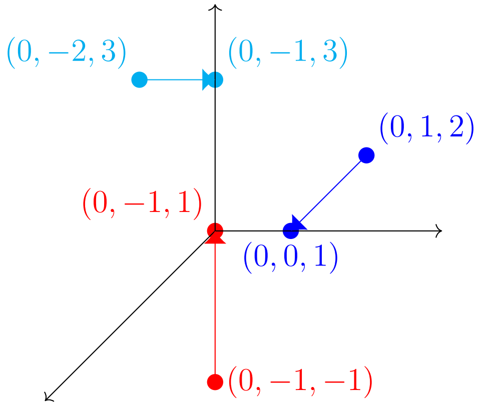


Figure 1.4: The tropical linear space L_p from 1.15, along with projections of three points not in L_p.

Blue Rule constructs a point $w \in \mathbb{R}^3 / \mathbb{R}\mathbf{1}$ whose first coordinate is

$$\min(\max(p_{12} - p_{12} + u_1, p_{12} - p_{23} + u_3), \max(p_{13} - p_{13} + u_1, p_{13} - p_{23} + u_2)).$$

Substituting in, we get the first coordinate of w as

$$w_1 = \min(\max(-2 + 2 + 0, -2 + 1 + 3), \max(-4 + 4 + 0, 0 - 1 - 2)) = \min(2, 0) = 0.$$

Similarly, we get $w_2 = -1$ and $w_3 = 3$. So the Blue Rule outputs the vector $(0, -1, 3)$.

The Red Rule constructs a vector v as follows. First, we begin with $v = (0, 0, 0)$. Next we take the set $\tau = \{1, 2, 3\}$ and compute $\min(p_{23} + u_1, p_{13} + u_2, p_{12} + u_3) = \min(-2 + 3, 0 + -2, -1 + 0) = -2$. *So the Red Rule redefines $v_2 = -1 - (-2) = 1$, and hence outputs the vector $v = (0, 1, 0)$, which by Theorem 1.18 satisfies $w = u + v$.*

Definition 1.20. *Let $v = (v_1, \ldots, v_n)$ be a real vector, and define the tropical linear functional $\bigoplus(-v_i) \odot x_i$. Let $\mathcal{H}$ be the tropical solution set of this linear functional: that is, $\mathcal{H}$ consists of all $x \in \mathbb{R}^n / \mathbb{R}\,\mathbf{1}$ such that the minimum of $\bigoplus(-v_i) \odot x_i$ is attained at least twice. We call any $\mathcal{H}$ obtained in this way a* tropical hyperplane.

Remark 1.21. *Let A be a tropical matrix of dimensions $(n-1) \times n$. Then the Stiefel tropical linear space of A is a tropical hyperplane. Furthermore, any tropical hyperplane is the Stiefel tropical linear space of such a tropical matrix A.*

1.3 Tropical convexity

We next review some basics of tropical convexity, again following [52, Chapters 5] and [63, Chapter 4]. In discussing tropical convexity, it is often convenient to work in *tropical projective space* $\mathbb{TP}^{n-1} = (\mathbb{T}^n \backslash (\infty, \infty, \ldots, \infty))/\mathbb{R}\mathbf{1}$, which contains the tropical projective torus $\mathbb{R}^n / \mathbb{R}\,\mathbf{1}$. Indeed, the tropical projective space $\mathbb{TP}^{n-1}$ is a compactification of the tropical projective torus, and the pair of spaces $(\mathbb{TP}^{n-1}, \mathbb{R}^n / \mathbb{R}\,\mathbf{1})$ is homeomorphic to the $(n-1)$-dimensional simplex and its interior [52, Proposition 5.3].

Given a finite collection $\mathcal{P}$ of points in $\mathbb{TP}^{n-1}$, we define their *tropical convex hull* or *tropical polytope* $P = \operatorname{tconv}(\mathcal{P})$ as the tropical semimodule spanned by these points, i.e.:

$$\operatorname{tconv}(\mathcal{P}) = \{\lambda_1 \odot p^{(1)} \oplus \cdots \oplus \lambda_s \odot p^{(s)} : \lambda_i \in \mathbb{T}, p^{(i)} \in \mathcal{P}\},$$

where $\lambda \odot p = (p_1 + \lambda, \ldots, p_n + \lambda)$ is tropical scalar multiplication. In fact, tropical linear spaces in $\mathbb{TP}^{n-1}$ can be viewed as tropical polytopes, spanned by points on the boundary of $\mathbb{TP}^{n-1}$ called their *cocircuits* [52, Proposition 10.33].

Let $P = \operatorname{tconv}(\mathcal{P}) \subseteq \mathbb{TP}^{n-1}$ be a tropical polytope. The *(tropical) type* of a point $\mathbf{x}$ in $\mathbb{TP}^{n-1}$ with respect to $\mathcal{P}$ is the collection of sets $S = (S_1, \ldots, S_n)$, where an index i is contained in S_j if

$$p_j^{(i)} - x_j = \min(p_1^{(i)} - x_1, \ldots, p_n^{(i)} - x_n).$$

The tropical polytope P consists of all points $\mathbf{x}$ whose type $S = (S_1, \ldots, S_n)$ has all S_i nonempty. Each collection of points with the same type in P is called a *cell*, and each cell with all S_i nonempty is a *polytrope*: a tropical polytope that is classically convex [53]. In this way all tropical polytopes have a decomposition into polytropes.

Example 1.22. *Consider the tropical polytope in $\mathbb{TP}^2$ spanned by vertices* $(0, 2, 2)$, $(0, 0, 1)$, $(0, 4, 0)$, *and* $(0, 5, 4)$. *Its cellular decomposition contains three full-dimensional polytropes, which are illustrated in Figure 1.5.*

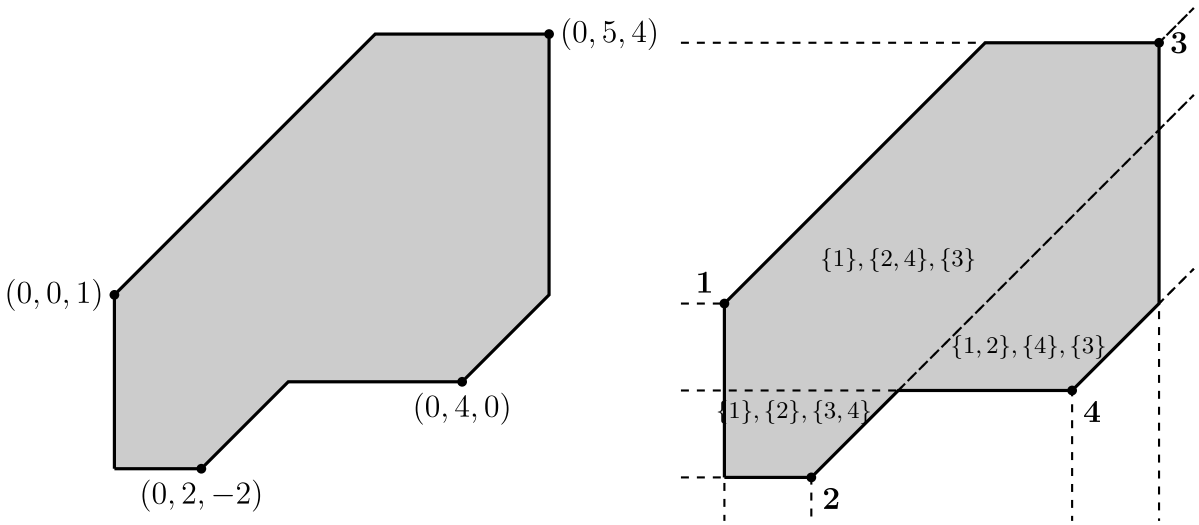


Figure 1.5: A 2-dimensional tropical polytope in $\mathbb{TP}^2$ spanned by four vertices, and its decomposition into three polytropes, labeled with their types.

Let $P = \text{tconv}(p^{(1)}, p^{(2)}, \ldots, p^{(s)})$ be a tropical polytope. There is a projection map $\pi_{\mathcal{P}}$ sending any point x to a closest point on the tropical polytope P in the tropical metric given in Definition 1.16:

$$\pi_{\mathcal{P}}(x) \;=\; \lambda_1 \odot p^{(1)} \,\oplus\, \lambda_2 \odot p^{(2)} \,\oplus\, \cdots \,\oplus\, \lambda_s \odot p^{(s)}, \quad \text{where } \lambda_k = \max(x - p^{(k)}). \qquad (1.3.1)$$

Viewing a tropical linear space as a tropical polytope spanned by its cocircuits, this projection map generalizes the Blue Rule [52, Proposition 10.77].

The lattice of integral points $\mathbb{Z}^n \subseteq \mathbb{R}^n$ induces a flag simplicial complex structure on $\mathbb{TP}^{n-1}$, with the quotients of lattice points in $\mathbb{Z}^n$ as the vertices and a 1-simplex between two lattice points if they are of tropical distance 1 apart, as illustrated in Figure 1.6 for $\mathbb{TP}^2$. This is called the *standard triangulation* of $\mathbb{TP}^{n-1}$. Given points $\mathcal{P} \subseteq \mathbb{Z}^n$, we call their tropical convex hull a *tropical lattice polytope*. The standard triangulation of $\mathbb{TP}^{n-1}$ descends to a standard triangulation of any tropical lattice polytope [51, Theorem 11].

1.4 Tropical geometry and phylogenetics

In this section we describe some of the tropical structures underlying the space of ultrametric trees. In keeping with Chapter 5, we switch to the max-plus tropical semiring for this section. In what follows, let $m \geq 2$ be a natural number and $e = \binom{m}{2}$. Our treatment of this subject largely follows [63, Section 4.3].

Definition 1.23. *A* dissimilarity map d *is a function* $d : [m] \times [m] \to \mathbb{R}_{\geq 0}$ *such that* $d(i,i) = 0$ *and* $d(i,j) = d(j,i) \geq 0$ *for each* $i,j \in [m]$. *If, furthermore, we have that*

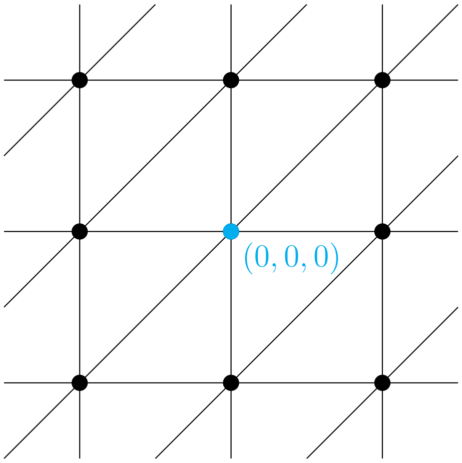


Figure 1.6: The standard triangulation of $\mathbb{TP}^2$, with the origin colored cyan.

$d(i,j) \leq d(i,k) + d(k,j)$ for all $i, j, k \in [m]$, we call d a metric*. Note that for convenience we often write d_{ij} for the term $d(i,j)$.*

We can represent a dissimilarity map d by an $m \times m$ matrix D whose (i,j)th entry is d_{ij}. Because D is symmetric and all diagonal entries are zero, there is a natural encoding of d as a vector in $\mathbb{R}^e = R^{\binom{m}{2}}$. The metrics we are most interested in come from phylogenetic trees:

Definition 1.24. *Let $T = (V, E)$ be a tree with m labeled leaves and no vertices of degree two. We call such a tree a* phylogenetic tree*.*

Definition 1.25. *Let T be a phylogenetic tree with m leaves labeled by the elements of $[m]$, and assign a length $\ell_e \in \mathbb{R}$ to each edge e of T. Let $d : [m] \times [m] \to \mathbb{R}$ be defined so that d_{ij} is the total length of the unique path from leaf i to leaf j. We call a function d obtained in this way a* tree distance*. If, furthermore, each entry of the distance matrix D is nonnegative, then d is in fact a metric. We call such a d a* tree metric*.*

As before, we can embed D as a vector into $\mathbb{R}^e$. Any tree distance differs from a tree metric by some scalar multiple of $\mathbf{1}$; hence the sets of tree distances and tree metrics coincide in the tropical projective torus.

Definition 1.26. *Let $d : [m] \times [m] \to \mathbb{R}_{\geq 0}$ be a metric which satisfies the following strengthening of the triangle inequality for each choice of $i, j, k \in [m]$:*

$$d(i,k) \leq \max(d(i,j), d(j,k)).$$

We call such a metric an ultrametric*, and denote by $\mathcal{U}_m$ the set of all ultrametrics in $\mathbb{R}^e/\mathbb{R}\mathbf{1}$.*

All ultrametrics can be derived from an *equidistant tree* with non-negative edge weights, where all leaves have the same distance to some distinguished root vertex. Furthermore, the tree metric of an equidistant tree with non-negative edge weights is an ultrametric; hence ultrametrics and equidistant trees with non-negative edge weights convey equivalent information.

Example 1.27. *Consider the two equidistant phylogenetic trees in Figure 1.7. In the first tree, the distance between leaves A and B is 4, as is the distance between leaves A and C, while the distance between B and C is 2. We can view the tree as the vector $v = (4, 4, 2)$ in $\mathbb{R}^{\binom{3}{2}}$, and similarly we view the second tree as $w = (2, 4, 4)$.*

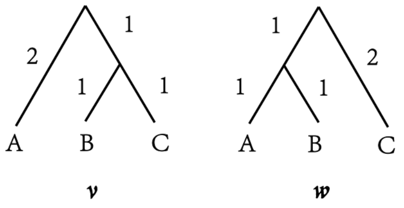


Figure 1.7: Two equidistant phylogenetic trees

Remark 1.28. *The tropical distance in Equation* (1.2.1) *between two ultrametric trees measures the range of disagreement between the two tree metrics. Consider the two phylogenetic trees $v = (4, 4, 2)$ and $w = (2, 4, 4)$ in Example 1.27. The largest disagreement between v and w in which tree v finds a longer distance between two leaves is* $\max\{v_i - w_i\} = 2$*, and the largest disagreement between v and w in which tree w shows a bigger distance between two leaves is* $\max\{w_j - v_j\} = 2$*. So $d_{tr}(v, w) = 2 + 2 = 4$.*

Let L_m denote the subspace of $\mathbb{R}^e$ defined by the linear equations $x_{ij} - x_{ik} + x_{jk} = 0$ for $1 \leq i < j < k \leq m$. The tropicalization $\operatorname{trop}(L_m) \subseteq \mathbb{R}^e / \mathbb{R}\mathbf{1}$ is the tropical linear space consisting of points $(v_{12}, v_{13}, \ldots, v_{m-1,m})$ such that $\max(v_{ij}, v_{ik}, v_{jk})$ is obtained at least twice for all triples $i, j, k \in [m]$. This is in fact a rephrasing of the ultrametric condition, meaning the space of ultrametrics is itself a tropical linear space:

Theorem 1.29. *The image of $\mathcal{U}_m$ in the tropical projective torus $\mathbb{R}^e / \mathbb{R}\mathbf{1}$ coincides with* $\operatorname{trop}(L_m)$.

1.5 Contributions in this dissertation

Having developed some basics of tropical geometry, we now state the main new results of this dissertation. We focus on the development of new algorithms for the computation of interesting objects in tropical geometry.

Chapter 2 is based on [17], which is joint work with Justin Chen and Sameera Vemulapalli. It will be published in the Journal of Symbolic Computation in May 2021. The chapter focuses on developing algorithms for computing intrinsic tropicalizations of various classes of curves. The notion of tropical variety in Definition 1.6 depends in an essential way on the particular embedding of the very affine variety $V(I)$ into the torus. Intrinsic tropicalizations provide a path toward removing this embedding dependence, but no general methods for computing intrinsic tropicalizations existed prior to our work, as the notion relies on an explicit $\mathbb{Z}$-basis for the unit group of the corresponding very affine variety. Chapter 2 provides effective methods for the computation of intrinsic tropicalizations for plane conics, rational normal curves, and elliptic curves.

Theorem 1.30. *Algorithm 2.26, Algorithm 2.34, and Algorithm 2.47 correctly compute $\mathbb{Z}$-bases for the unit groups of very affine plane conics, rational normal curves, and elliptic curves, respectively.*

Example 1.31. *Let $\overline{C}$ be the conic defined by $f = (1+t)x^2 + (1+t)y^2 + (1+t)z^2 - (2+2t+t^2)xy - (2+2t+t^2)yz - (2+2t+t^2)xz$ over the field of Puiseux series in t over $\mathbb{C}$. Consider the very affine curve C given by intersecting with the canonical torus. We note that the tropicalization of f is simply the tropical line $0 \oplus x \oplus y$ shown in the left in Figure 1.8. The intrinsic tropicalization has the following snowflake structure typical of a generic tropical conic as shown in the right in Figure 1.8.*

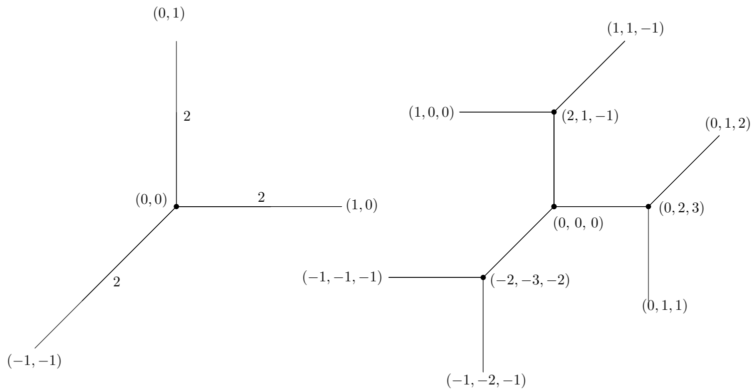


Figure 1.8: The tropicalization and intrinsic tropicalization respectively of the conic in Example 1.31, with coordinates and nontrivial multiplicities labeled.

In Chapter 3, which is based on joint work with Paul Görlach and Yue Ren [37] submitted to Computational Complexity, we describe new methods for the computation of zero-dimensional tropical varieties. A number of algorithms for the computation of general tropical varieties rely on reductions to the zero-dimensional case. We describe a new approach for the computation of such a base case, by computing projections of the zero-dimensional tropical variety onto 1-dimensional lines and using these projections to reconstruct the original variety. Recall that the tropical variety of an ideal in a single variable can be read straightforwardly from the slopes of the corresponding Newton polygon.

Theorem 1.32. *Let K be a field with non-trivial valuation ν and let $I \subseteq K[\mathbf{x}^{\pm}]$ be a zero-dimensional ideal in triangular form. Then Algorithm 3.9 correctly computes the tropicalization* $\mathrm{trop}(I)$ *using polynomially-many arithmetic operations in K and $\mathbb{Q}$.*

Example 1.33. *Consider $K = \mathbb{Q}$ equipped with the 2-adic valuation and the ideal*

$$I = \langle \underbrace{2x_3^4 + x_3^3 + x_3^2 + x_3 + 2}_{=:g_3}, x_2 - \underbrace{2x_3}_{=:f_2}, x_1 - \underbrace{4x_3}_{=:f_1} \rangle \subseteq K[x_1^{\pm}, x_2^{\pm}, x_3^{\pm}].$$

Using the Newton polygons in Figure 1.9, we can reconstruct the tropicalization of I as

$$\mathrm{trop}(I) = \mathrm{trop}(I)_{\{1,2,3\}} = \{(3,2,1), (\mathbf{2},\mathbf{1},\mathbf{0}), (1,0,-1)\}.$$

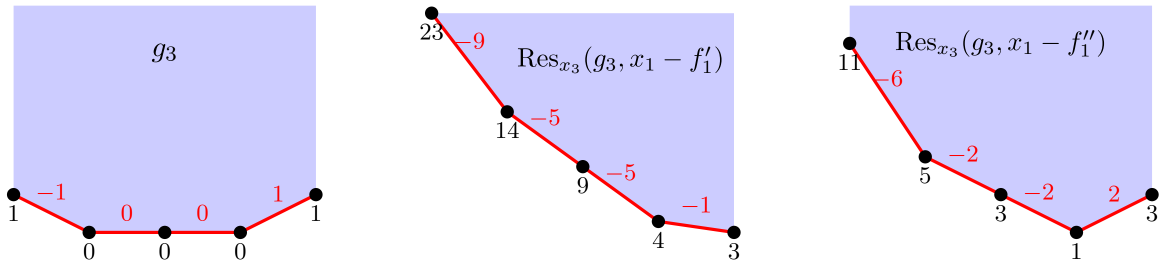

g_3
1
-1
0
0
0
0
0
1
1
$\mathrm{Res}_{x_3}(g_3, x_1 - f_1')$
23
-9
14
-5
9
-5
4
-1
3
$\mathrm{Res}_{x_3}(g_3, x_1 - f_1'')$
11
-6
5
-2
3
-2
1
2
3

Figure 1.9: Newton polygons of g_3 and other polynomials relevant to Example 1.33. Below each vertex is its height, above each edge is its slope.

In Chapter 4, we consider certain subsets of the affine building of SL_d called min-convex hulls. This chapter is based on [91], which will be published in Discrete & Computational Complexity. Min-convex hulls were originally introduced by Faltings [27], who noted that their points in $\mathcal{B}_d$ correspond to smooth irreducible components of the special fiber of certain regular schemes $M(\Gamma)$ called *Mustafin varieties* or *Deligne schemes* over the spectrum of a discrete valuation ring. Joswig, Sturmfels, and Yu showed that min-convex hulls were isomorphic to tropical polytopes in some tropical projective torus and provided a partial algorithm for the computation of min-convex hulls in this way [51]. However, their algorithm required a certain structure called an *enveloping membrane* containing the min-convex hull of interest. We describe effective algorithms for the computation of enveloping membranes, yielding the first time-bounded complexity algorithms to calculate min-convex hulls as well.

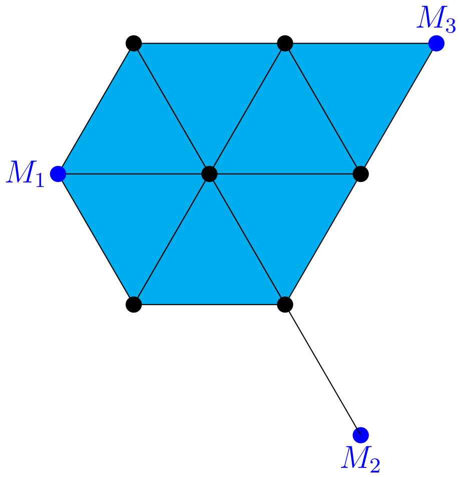


Figure 1.10: The convex hull of the matrices M_1, M_2, and M_3 in $\mathcal{B}_3$.

Theorem 1.34. *Let $M_1, \ldots, M_s \in K^{d\times d}$ represent lattices $\Lambda_1, \ldots, \Lambda_s \in \mathcal{B}_d$. Then Algorithm 4.29 correctly computes an enveloping membrane for* $\operatorname{conv}(\Lambda_1, \ldots, \Lambda_s)$*, whose representative matrix M has at most $d \cdot 2^d \cdot (d!)^{s-3}$ columns.*

Example 1.35. *Let $K = \mathbb{Q}_5$ and consider the matrices*

$$M_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, M_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{5} & 0 \\ 0 & 0 & \frac{1}{125} \end{pmatrix}, M_3 = \begin{pmatrix} 5 & 625 & 150 \\ 0 & 25 & 1 \\ 0 & 0 & \frac{1}{5} \end{pmatrix}.$$

Let M be the matrix

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

The membrane $[M]$ contains the min-convex hulls of M_1, M_2, and M_3. We can use $[M]$ to compute their min-convex hull as shown in Figure 4.4.

In Chapter 5 we explore a tropical analogue of principal component analysis for phylogenetics. The chapter is based on joint work with Ruriko Yoshida and Xu Zhang [90] which is published in the Bulletin of Mathematical Biology, as well as another paper with Ruriko Yoshida and Robert Page [89] which will be published in Bioinformatics. Advances in genome sequencing have produced large datasets of phylogenetic trees. These datasets are ill-suited for analysis with classical statistical tools like principal component analysis as the underlying data spaces are non-Euclidean. Conversely, as described in Section 1.4, the space of ultrametrics has an intrinsic tropical structure as a tropical linear space. We describe two tropical analogues of principal component analysis using tropical linear spaces and tropical polytopes. We characterize an existing notion called the tropical volume in terms of tropical principal hyperplanes and show that the polytropal decomposition of a tropical polytope spanned by ultrametric trees preserves tree topologies. We also describe methods for computing tropical principal components and conduct experiments with real-world phylogenetic datasets.

Theorem 1.36. *Let $D^{(1)}, \dots, D^{(e)}$ be a collection of e points in $\mathbb{R}^e / \mathbb{R}\mathbf{1}$. Then any best-fit hyperplane attains a distance from the e points equal to their tropical volume, and Algorithm 5.12 attains such a hyperplane with high probability.*

Theorem 1.37. *Let $\mathcal{P} = \operatorname{tconv}(D^{(1)}, \dots, D^{(t)})$ be a tropical polytope spanned by ultrametrics, as computed for example by Algorithm 5.24. Then the polytropal decomposition of $\mathcal{P}$ refines the decomposition of $\mathcal{P}$ into different tree topologies.*

Example 1.38. *The genomes of eight species of protozoa in the Apicomplexa phylum were sequenced and converted into a dataset of 252 equidistant trees by [57]. These equidistant trees lie in a non-Euclidean 28-dimensional space. We computed a two-dimensional tropical principal polytope minimizing the sum of tropical distances of the datapoints and plotted their projections in Figure 1.11. Note that different polytropal components of the principal polytope correspond to particular tree topologies, which are color-coded.*

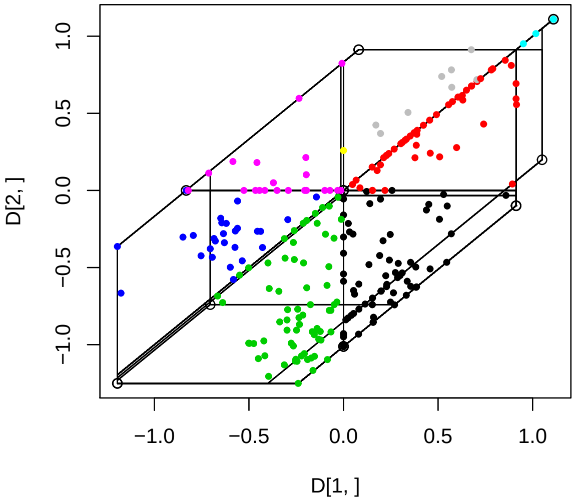


Figure 1.11: Projected points in the tropical principal polytope.

Chapter 2

Intrinsic tropicalizations of curves

The original material in this chapter is joint work with Justin Chen and Sameera Vemulapalli. The chapter appeared in modified form under the title “Computing unit groups of curves” in the Journal of Symbolic Computation [17].

2.1 Introduction

Among the invariants of a commutative ring, the group of units is one of the most fundamental. However, explicit computation of this group is difficult, and even its structure remains mysterious in general [34]. To date, most progress has centered on rings of integers of algebraic number fields, or localizations thereof, driven by a need for practical algorithms in computational number theory [18]. These results rely fundamentally on Dirichlet’s unit theorem, which describes the group of units, modulo torsion, of a number field as a free abelian group of finite rank specified by simple invariants of the number field.

An analogous theorem, proved independently by Rosenlicht and Samuel [76, 78], states that for a finitely generated domain over an algebraically closed field, the group of units, modulo scalars, is free abelian of finite rank. In contrast to the number field case, no formula for the rank is known. However, there is still interest in understanding the unit group: for a very affine variety, generators for the unit group of the coordinate ring yield an embedding of the variety into its so-called *intrinsic torus* [63, Chapter 6.4]. In tropical geometry, this embedding of a very affine variety into its intrinsic torus realizes its *intrinsic tropicalization*, from which all other tropicalizations can be recovered. Computing the intrinsic tropicalization is difficult, though, because one must first compute the unit group.

In this work we describe effective methods for computing unit groups of smooth very affine curves of low genus. Our methods rely on divisor theory for projective varieties: we embed the unit group of a very affine variety into the Weil divisor group of the projective closure, and study the cokernel of this embedding as a subgroup of the divisor class group. This allows us to give algorithms for computing unit groups of rational normal curves and elliptic curves:

Theorem 2.1. *Let $\overline{C} \subseteq \mathbb{P}^n_k$ be a rational normal curve over an algebraically closed field k, given parametrically as the image of a map $\mathbb{P}^1_k \hookrightarrow \mathbb{P}^n_k$. Let $C := \overline{C} \cap \mathbb{T}^n$ be the corresponding very affine curve, with coordinate ring R. Then Algorithm 2.34 correctly computes a $\mathbb{Z}$-basis of R^*/k^*.*

Theorem 2.2. *Let $k = \overline{\mathbb{Q}}$, let $\overline{E} \subseteq \mathbb{P}^2_k$ be an elliptic curve, and let $E := \overline{E} \cap \mathbb{T}^2$ be the corresponding very affine elliptic curve with coordinate ring R. Then Algorithm 2.47 correctly computes a $\mathbb{Z}$-basis of R^*/k^*.*

We briefly describe the structure of the chapter. The basics of unit groups and intrinsic tropicalizations are discussed in Section 2.2. In Section 2.3 we interpret the problem of computing units via the geometry of boundary divisors, and give an algorithm for interpolating divisors of rational functions to Laurent polynomials. We consider two families of plane curves in Section 2.4, namely Fermat curves and conics. Section 2.5 deals with rational normal curves in parametric form. Finally, we discuss elliptic curves in Section 2.6.

Many of our algorithms have been implemented in Macaulay2 [38], Singular [20], or Sage [77]. Our code for the examples in this chapter can be found at:

`https://github.com/leonyz/unitgroups/`

2.2 Background

We begin by stating the problem in a general setting. Let k be an algebraically closed field, and let R be a finitely generated k-algebra which is a domain. The inclusion $k \subseteq R$ induces a short exact sequence of multiplicative abelian groups

$$1 \longrightarrow k^* \longrightarrow R^* \longrightarrow R^*/k^* \longrightarrow 1 \tag{2.2.1}$$

Our goal is to compute, as explicitly as possible, the group R^*/k^*. Although this may seem to be a purely algebraic problem, the key to progress is to use insights from geometry, particularly divisor theory on projective varieties. Thus, writing $R = k[x_1, \ldots, x_n]/I$ as a quotient of a polynomial ring by a prime ideal I, set $X := \operatorname{Spec} R \subseteq \mathbb{A}^n_k$, the affine variety corresponding to R, and let $\overline{X} \subseteq \mathbb{P}^n_k$ denote the projective closure of X in projective n-space. Write $\partial X := \overline{X} \setminus X = \overline{X} \cap V(x_0)$ for the boundary of $\overline{X}$, which is the intersection of $\overline{X}$ with the hyperplane at infinity in $\mathbb{P}^n_k$.

The main point is that a unit in R corresponds, via homogenization, exactly to a rational function on $\overline{X}$ which has zeros and poles only on ∂X. To be precise:

Lemma 2.3. *With notation as above, let $R \to \overline{R}$ be the homogenization map $f \mapsto \overline{f} := x_0^{\deg f} f(\frac{x_i}{x_0})$. Then:*

i) For any $f, g \in R$, $\overline{fg} = \overline{f}\,\overline{g}$, and

ii) $f \in R^*$ if and only if $V(\overline{f}) \cap \overline{X} \subseteq \partial X$.

Proof. First, note that dehomogenization is evaluation at $x_0 = 1$, hence is a ring map with kernel $(x_0 - 1)$. As the kernel contains no nonzero homogeneous elements, it follows that if f_1, f_2 are homogeneous of the same degree with the same dehomogenization, then $f_1 = f_2$.

i) Since $\overline{fg}$ and $\overline{f}\,\overline{g}$ are both homogeneous of the same degree and dehomogenize to fg, by the above reasoning they must be equal.

ii) As $\partial X = \overline{X} \cap V(x_0)$, it suffices to show that $f \in R^*$ if and only if $V(\overline{f}) \cap \overline{X} \subseteq V(x_0)$. If $g_1, \ldots, g_r$ is a Gröbner basis for the defining ideal I of X with respect to a term order refining total degree, then $\overline{X}$ has defining ideal $(\overline{g_1}, \ldots, \overline{g_r})$ [25, Prop. 15.31b]. It thus suffices to show $1 \in (f, g_1, \ldots, g_r)$ if and only if $x_0 \in \sqrt{(\overline{f}, \overline{g_1}, \ldots, \overline{g_r})}$. The "if" direction follows by dehomogenizing. For the "only if" direction, pick h with $1 - fh \in I$. Then $\overline{1 - fh} \in (\overline{g_1}, \ldots, \overline{g_r})$. But $\overline{1 - fh} = x_0^d - \overline{fh}$, where $d = \deg(fh)$, as both sides are homogeneous and dehomogenize to $1 - fh$. Applying (i) gives $x_0^d \in (\overline{f}, \overline{g_1}, \ldots, \overline{g_r})$ as desired. □

Suppose now that $\overline{X}$ is normal, and write $\mathrm{Div}(\overline{X})$ (resp. $\mathrm{Cl}(\overline{X})$) for the group of Weil divisors (resp. the divisor class group) on $\overline{X}$. Let $\mathrm{Div}^0(\overline{X})$ (resp. $\mathrm{Cl}^0(\overline{X})$) denote the subgroup of divisors (resp. divisor classes) of degree zero.

Definition 2.4. *We define*

$$\mathrm{Div}^0_\partial(\overline{X}) := \left\{ \sum_{\text{finite}} a_i P_i \;\middle|\; P_i \text{ component of } \partial X, a_i \in \mathbb{Z}, \sum a_i = 0 \right\} \subseteq \mathrm{Div}^0(\overline{X})$$

i.e. the subgroup of $\mathrm{Div}^0(\overline{X})$ *supported on* ∂X *(which makes sense as* ∂X *is codimension 1 in* $\overline{X}$*).*

By Lemma 2.3(i), we have a composition of abelian group homomorphisms

$$\begin{gathered} R^* \to \mathrm{Frac}(\overline{R})^* \to \mathrm{Div}^0(\overline{X}) \\ u \mapsto \overline{u}, f \mapsto \mathrm{div}(f) \end{gathered} \tag{2.2.2}$$

consisting of homogenization followed by the divisor map. Lemma 2.3(ii) shows that the image of R^* is contained in $\mathrm{Div}^0_\partial(\overline{X})$, so there is an induced map $\widetilde{\phi} : R^* \to \mathrm{Div}^0_\partial(\overline{X})$. Now $\ker \widetilde{\phi}$ consists of functions which have no zeros or poles anywhere on $\overline{X}$. Such an element must be a scalar, i.e. lies in k^*. Thus we have a map $\phi : R^*/k^* \hookrightarrow \mathrm{Div}^0_\partial(\overline{X})$.

Putting the above reasoning together yields a classical theorem of Rosenlicht and Samuel [76, 78] on the structure of the unit group:

Theorem 2.5. *Let k be an algebraically closed field, and let R be a finitely generated k-algebra that is a domain. Then R^*/k^* is a finitely generated free abelian group.*

Proof. Let $\overline{R}$ be the homogenization of R with respect to some new variable x_0. If $\overline{X} = \mathrm{Proj}(\overline{R})$ is normal, then the reasoning above shows that R^*/k^* embeds in the finitely generated free abelian group $\mathrm{Div}^0_\partial(\overline{X})$, and subgroups of finitely generated free abelian groups are again finitely generated free abelian.

If $\overline{X}$ is not normal, let $\widetilde{\overline{X}}$ be the normalization of $\overline{X}$. The normalization map $\widetilde{\overline{X}} \xrightarrow{\eta} \overline{X}$ identifies $\eta^{-1}(X)$ with $\operatorname{Spec}(\widetilde{R})$, where $\widetilde{R}$ is the integral closure of R in its fraction field. This gives an inclusion map $R^*/k^* \hookrightarrow (\widetilde{R})^*/k^*$. As $(\widetilde{R})^*/k^*$ is finitely generated free abelian by the previous case, R^*/k^* is as well. □

Remark 2.6. *Note that the unit group of the coordinate ring of a projective variety is trivial to compute: indeed, in this case $\overline{R}^* = k^*$, as any positively graded domain has units concentrated in degree 0. Thus Theorem 2.5 is only interesting for rings which are not positively graded.*

Remark 2.7. *The assumptions in Theorem 2.5 are necessary: if k is not algebraically closed, then the unit group modulo scalar units may have torsion, i.e. roots of unity. If R is not a domain, then R^*/k^* need not be $\mathbb{Z}$-free or even finitely generated: e.g. for $R = k[x]/\langle x^2\rangle$, one has $R^*/k^* \cong (k, +)$ is the additive group of the field.*

However, Theorem 2.5 still holds if R is reduced: if $p_1, \ldots, p_m$ are the minimal primes of R, then the diagonal embedding $\Delta : R \hookrightarrow \prod_{i=1}^m R/p_i$ gives $R^* \hookrightarrow \prod_{i=1}^m (R/p_i)^*$, and since $\Delta^{-1}(\prod_{i=1}^m k^*) = k^*$ (as zeros/poles must appear on some component if they appear at all), one has that $R^*/k^* \hookrightarrow \prod_{i=1}^m (R/p_i)^*/k^*$ is a subgroup of a finite product of free abelian groups.

Remark 2.8. *In the setting of Theorem 2.5, freeness of R^*/k^* implies the exact sequence (2.2.1) splits, i.e. $R^* \cong k^* \oplus R^*/k^*$. Thus if we understand R^*/k^*, then we also understand R^*.*

Intrinsic tropicalization

We now discuss some motivation for computing unit groups coming from tropical geometry [85], following the presentation in [63]. Recall that a variety X is said to be *very affine* if X admits a closed embedding into an algebraic torus $\mathbb{T}$. Intuitively, a subvariety of $\mathbb{P}^m$ is affine if it misses a coordinate hyperplane, and very affine if it misses all coordinate hyperplanes. Algebraically, this means that the coordinate ring R of X is (isomorphic to) a quotient of a Laurent polynomial ring $k[x_1^{\pm}, \ldots, x_m^{\pm}]$. We note that given a very affine variety $X \subseteq \mathbb{T}^n$, one can take its projective closure $\overline{X} \subseteq \mathbb{P}^n$ with boundary $\partial X := \overline{X} \setminus X = \overline{X} \cap V(x_0 \cdots x_n)$, and the above discussion (cf. Lemma 2.3, Definition 2.4) carries over to this setting.

In general, there are many different closed embeddings of X into tori $\mathbb{T}^m$ for various m. To remove the dependence on the choice of embedding, one must choose a "natural" embedding of X into a fixed torus. As it turns out, the right object to consider is the so-called *intrinsic torus* of X, which is by definition [63, Definition 6.4.2]

$$\mathbb{T}_{in} := \operatorname{Hom}_{\mathbb{Z}}(R^*/k^*, k^*).$$

Note that by Theorem 2.5, R^*/k^* is free abelian, so the Hom group is isomorphic to a product of copies of k^*, which is an algebraic torus over k. A $\mathbb{Z}$-basis $f_1, \ldots, f_n$ of R^*/k^* gives rise to an embedding $i : X \hookrightarrow \mathbb{T}_{in}$, via $x \mapsto (f_1(x), \ldots, f_n(x))$. With such a choice of basis, the importance of the intrinsic torus is immediate from the following "pseudo-universal" property [63, Proposition 6.4.4]: for every closed embedding $j\colon X \hookrightarrow \mathbb{T}^m$ of X into a torus, there is a map of tori $\varphi : \mathbb{T}_{in} \to \mathbb{T}^m$ given by Laurent monomials (which need not be an embedding) such that the following diagram commutes:

$$\begin{tikzcd}
X \arrow[r, hook, "i"] \arrow[dr, hook, "j"'] & \mathbb{T}_{in} \arrow[d, "\varphi"] \\
& \mathbb{T}^m
\end{tikzcd}$$

It is a basic task in tropical geometry to tropicalize a very affine variety with respect to a particular embedding in a torus. From a foundational viewpoint, it is desirable to have an *intrinsic tropicalization*, with respect to the intrinsic torus, so that the tropicalization depends only on the very affine variety X and not the specific embedding $X \hookrightarrow \mathbb{T}^m$. Furthermore, in the setup of the commutative diagram above, the tropicalization of X embedded in $\mathbb{T}^m$ is given by the image of the intrinsic tropicalization under the affine map $\operatorname{trop}(\varphi)$. Hence any other tropicalization of X can be recovered from the intrinsic tropicalization.

However, from a computational standpoint, the very affine variety is most often described by its ideal in a fixed embedding. To obtain an intrinsic tropicalization one must be able to compute the defining ideal of the very affine variety in its intrinsic torus; the key to doing so is to first compute a basis of R^*/k^*. Although an embedding into the intrinsic torus depends on our choice of basis for R^*/k^*, we nevertheless often speak of *the* intrinsic embedding into the intrinsic torus.

2.3 General results on varieties

In this section we outline our approach to the problem of computing unit groups, via class groups. We retain the setup from the previous section: let X be a normal very affine variety over a field $k = \overline{k}$, with coordinate ring R, projective closure $\overline{X} \subseteq \mathbb{P}^n_k$, and boundary $\partial X = \overline{X} \setminus X$.

Definition 2.9. *Define* $\mathrm{Cl}^0_\partial(\overline{X})$ *to be the cokernel of the embedding* $R^*/k^* \xhookrightarrow{\phi} \mathrm{Div}^0_\partial(\overline{X})$.

By definition, there is a short exact sequence of abelian groups

$$1 \longrightarrow R^*/k^* \xrightarrow{\phi} \mathrm{Div}^0_\partial(\overline{X}) \longrightarrow \mathrm{Cl}^0_\partial(\overline{X}) \longrightarrow 0 \tag{2.3.1}$$

Corollary 2.10. *Let r be the number of divisorial components of ∂X. Then* $\operatorname{rank} R^*/k^* \leq r-1$, *with equality if and only if* $\mathrm{Cl}^0_\partial(\overline{X})$ *is torsion.*

Proof. The subgroup $\mathrm{Div}_\partial(\overline{X})$ of $\mathrm{Div}(\overline{X})$ (consisting of Weil divisors supported on ∂X) is a free group of rank r, and the degree 0 condition implies $\mathrm{Div}^0_\partial(\overline{X})$ is a free subgroup of rank $r-1$. □

Corollary 2.11. *If C is a very affine curve over k with coordinate ring R, with projective closure $\overline{C} \subseteq \mathbb{P}^n_k$ of degree d, then* $\operatorname{rank} R^*/k^* \leq (n+1)d - 1$.

Proof. As C is a curve, the divisorial components of ∂C are just the (closed) points of ∂C. Since C is very affine, the boundary ∂C consists of the intersections of $\overline{C}$ with each of the $n+1$ coordinate hyperplanes in $\mathbb{P}^n_k$. Then $\deg \overline{C} = d$ implies ∂C consists of at most $(n+1)d$ points, and the result follows from Corollary 2.10. □

Theorem 2.5 tells us that the structure of the unit group – as an *abstract* group – is as nice as possible. However, we need more information about the other groups in (2.3.1) to explicitly give generators for R^*/k^*. The following basic, but crucial, point states that all relations in $\mathrm{Cl}^0_\partial(\overline{X})$ are "geometric", in the sense that they come from the class group of $\overline{X}$.

Proposition 2.12. $\mathrm{Cl}^0_\partial(\overline{X})$ *is a subgroup of* $\mathrm{Cl}^0(\overline{X})$.

Proof. It suffices to show that the following composite map is injective:

$$\mathrm{Cl}^0_\partial(\overline{X}) \cong \mathrm{Div}^0_\partial(\overline{X})/(R^*/k^*) \overset{\alpha}{\hookrightarrow} \mathrm{Div}^0(\overline{X})/(R^*/k^*) \xrightarrow{\beta} \mathrm{Cl}^0(\overline{X})$$

i.e. $\mathrm{Im}(\alpha) \cap \ker(\beta) = 0$. But this follows since $\ker(\beta) = \mathrm{Frac}(\overline{R})^*/(R^*/k^*)$, and $\mathrm{Frac}(\overline{R})^* \cap \mathrm{Div}^0_\partial(\overline{X}) = R^*/k^*$, as a rational function on $\overline{X}$ supported only on ∂X is a unit on X. □

Remark 2.13. *Recall that the class group of the ring of integers of a number field is finite. If a similar result held in our setting, Corollary 2.10 would give an explicit description for the rank of R^*/k^*. Unfortunately, of course,* $\mathrm{Cl}(\overline{X})$ *need not be so well-behaved in general.*

In general, our approach to computing R^*/k^* via (2.3.1) proceeds in three parts:

Question 2.14. *What are generators of the image of R^*/k^* in* $\mathrm{Div}^0_\partial(\overline{X})$*?*

Question 2.15. *Given $D \in \mathrm{Div}^0_\partial(\overline{X})$ that is in the image of R^*/k^*, can we find polynomials f, g such that $f/g \in R^*/k^*$ is mapped to D (under the inclusion $R^* \subseteq$* $\mathrm{Frac}(\overline{R})^*$*)?*

Question 2.16. *Given an element of R^*/k^* expressed as a rational function as in Question 2.15, can we find a representative for it in R?*

Question 2.14 is precisely computing $\operatorname{im}\big(R^*/k^* \to \mathrm{Div}^0_\partial(\overline{X})\big) = \ker\big(\mathrm{Div}^0_\partial(\overline{X}) \to \mathrm{Cl}^0_\partial(\overline{X})\big)$. By Proposition 2.12, this also equals $\ker\big(\mathrm{Div}^0_\partial(\overline{X}) \to \mathrm{Cl}^0(\overline{X})\big)$. Thus to solve Question 2.14, one needs control over $\mathrm{Cl}^0(\overline{X})$, which is specific to the variety under consideration.

Question 2.15 is the first half of pulling back along (2.2.2), namely lifting $\mathrm{Frac}(\overline{R})^* \to \mathrm{Div}^0(\overline{X})$. This is an interpolation problem, and also depends on the variety in question.

Question 2.16 is the second half of pulling back along (2.2.2), namely lifting $R^* \to \mathrm{Frac}(\overline{R})^*$. This can in fact be easily solved with Gröbner bases, which we now show (throughout the chapter, we always work with a fixed term order refining total degree). Note that arguments involving Gröbner bases over polynomial rings can be adapted to Laurent polynomial rings by identifying the rings $k[x_1^{\pm},\dots,x_n^{\pm}] \cong k[x_1,\dots,x_n,t]/\langle tx_1\dots x_n - 1\rangle$ (alternatively, one can avoid adjoining an extra variable t and quotienting, by saturating with the product of the variables).

Algorithm 2.17 (Clearing denominators).

Require: $f, g \in k[x_1^{\pm 1},\dots,x_n^{\pm 1}]$, $I = \langle \phi_1,\dots,\phi_m\rangle \subseteq k[x_1^{\pm 1},\dots,x_n^{\pm 1}]$

Ensure: *$h \in k[x_1^{\pm 1},\dots,x_n^{\pm 1}]$ with $f - gh \in I$ if such an h exists, or* ***false*** *otherwise*

```
J ← I + ⟨g⟩
G ← GröbnerBasis(J)
if f ∉ ideal(G) then
    return false
end if
(C_0, ..., C_m) ← a vector over k[x_1^{±1}, ..., x_n^{±1}] such that f = C_0 g + C_1 φ_1 + ... + C_m φ_m
return C_0
```

Lemma 2.18. *For $f, g \in R = k[x_1^{\pm 1},\dots,x_n^{\pm 1}]/I$, Algorithm 2.17 correctly determines whether there exists $h \in R$ such that $f = gh$, and returns such an h if it exists.*

Proof. A standard Gröbner basis argument checks whether $f \in J$ and, if so, finds such a vector $C = (C_0,\dots,C_m)$ as above. Note that $f \in J$ if and only if there exists h such that $f - gh \in I$, so that $f = gh \in R$. □

Algorithm 2.19 (Testing units).

Require: $h \in k[x_1^{\pm 1},\dots,x_n^{\pm 1}]$, $I = \langle \phi_1,\dots,\phi_m\rangle \subseteq k[x_1^{\pm 1},\dots,x_n^{\pm 1}]$

Ensure: ***true*** *if $h \in (k[x_1^{\pm 1},\dots,x_n^{\pm 1}]/I)^*$, or* ***false*** *otherwise*

```
J ← I + ⟨h⟩
G ← GröbnerBasis(J)
if 1 ∈ ideal(G) then
    return true
end if
return false
```

Lemma 2.20. *For $h \in R = k[x_1^{\pm 1},\dots,x_n^{\pm 1}]/I$, Algorithm 2.19 correctly tests if h is a unit in R.*

Proof. A standard Gröbner basis argument checks whether $1 \in J$. Note that $1 \in J = I + \langle h\rangle \subseteq k[x_1^{\pm 1},\dots,x_n^{\pm 1}]$ if and only if $h \in (k[x_1^{\pm 1},\dots,x_n^{\pm 1}]/I)^*$. □

Algorithm 2.21 (Computing preimages of $R^* \to \mathrm{Frac}(\overline{R})^*$).

Require: *$\overline{f}, \overline{g} \in k[x_0,\dots,x_n]$ homogeneous, $\overline{f}/\overline{g} \in \mathrm{Frac}(\overline{R})^*$ and $I = \langle \phi_1,\dots,\phi_m\rangle \subseteq k[x_1^{\pm 1},\dots,x_n^{\pm 1}]$*

Ensure: *$h \in k[x_1^{\pm 1},\dots,x_n^{\pm 1}]$ such that $h = \overline{f}/\overline{g}$ in $\mathrm{Frac}(\overline{R})^*$ (via the inclusion $R^* \subseteq \mathrm{Frac}(\overline{R})^*$) if such an h exists, or* ***false*** *otherwise*

```
$f \leftarrow \overline{f}(1, x_1, \ldots, x_n)$
$g \leftarrow \overline{g}(1, x_1, \ldots, x_n)$
**if** *Algorithm 2.17*$(f, g, I) =$ **false then**
    **return false**
**else**
    $h \leftarrow$ *Algorithm* $2.17(f, g, I)$
    **if** *Algorithm 2.19*$(h, I) =$ **true then**
        **return** $h$
    **else**
        **return false**
    **end if**
**end if**
```

Lemma 2.22. *Let X be a very affine variety with coordinate ring R equal to $k[x_1^{\pm 1}, \ldots, x_n^{\pm 1}]/\langle \phi_1, \ldots, \phi_m \rangle$. Let $\overline{f}$ and $\overline{g}$ be homogeneous polynomials in $k[x_0, \ldots, x_n]$, and $f, g \in k[x_1, \ldots, x_n]$ their dehomogenizations with respect to x_0. Given a rational function $\overline{f}/\overline{g} \in \mathrm{Frac}(\overline{R})^*$, Algorithm 2.21 correctly decides whether $f/g \in R^*$ (via the inclusion $R^* \subseteq \mathrm{Frac}(\overline{R})^*$), and if so, computes a representative $h \in k[x_1^{\pm}, \ldots, x_n^{\pm}]$ for f/g.*

Proof. If $\overline{f}/\overline{g} \in R^*$ then there must exist a Laurent polynomial $h \in R^*$ such that $\overline{f}/\overline{g} = \overline{h}$ in $\mathrm{Frac}(\overline{R})^*$, where $\overline{h}$ is the homogenization of h with respect to x_0. Thus $\overline{f} - \overline{g}\overline{h} = 0$ in $\mathrm{Frac}(\overline{R})^*$, so $f - gh \in I$. Since $h \in R^*$, Algorithm 2.19 will verify that h is a unit, and Algorithm 2.21 will return h.

Now assume that $\overline{f}/\overline{g} \notin R^*$. Then Algorithm 2.21 will return false unless Algorithm 2.17 returns some $h \in R^*$ such that $f - gh$. Suppose this occurs. By homogenizing, we see that $\overline{f} = \overline{g}\overline{h}$ in $\overline{R}$ and $\overline{f}/\overline{g} = \overline{h}$ in $\mathrm{Frac}(\overline{R})^*$, which is a contradiction. □

2.4 Fermat curves and plane conics

We now consider two families of plane curves: Fermat curves and conics. These serve as our first two classes of examples for the general problem of computing unit groups.

Fermat curves

We first illustrate an elementary way of constructing units in quotients of Laurent rings:

Lemma 2.23. *Let $T := k[x_1^{\pm 1}, \ldots, x_d^{\pm 1}]$ be a Laurent polynomial ring, $I \subseteq T$ an ideal, $u \in T$ a monomial, $a \in k^*$, and $f \in I$. If there exist $g, h \in T$ with $f + au = gh$, then $\overline{g}, \overline{h}$ are units in $R := T/I$.*

Proof. Note that u is a unit in T (being monomial), so $a\overline{u}$ is a unit in R. Since $\overline{g}\overline{h} = a\overline{u} \in R^*$, we have that $\overline{g}$ and $\overline{h}$ are also units in R. □

Example 2.24 (Fermat curves)**.** *Consider the family of Fermat curves, which are plane curves in* $\mathbb{P}^2 = \mathrm{Proj}(k[x, y, z])$ *defined by equations of the form* $x^d + y^d = z^d$*, for* $d \in \mathbb{N}$*. For a fixed degree* d*, we have* $\overline{C} \coloneqq V(x^d + y^d - z^d) \subseteq \mathbb{P}^2$ *with homogeneous coordinate ring* $\overline{R} \coloneqq \mathbb{C}[x, y, z]/\langle x^d+y^d-z^d\rangle$*. Dehomogenizing with respect to* z *and intersecting with the torus in* $\mathbb{A}^2$ *gives a very affine Fermat curve* C *with coordinate ring* $R = \mathbb{C}[x^{\pm 1}, y^{\pm 1}]/\langle x^d + y^d - 1\rangle$*.*

We will use (2.3.1) *and Lemma 2.23 to show that the unit group* R^*/k^* *has* $3d - 1$ *independent elements. By Corollary 2.11,* $\operatorname{rank} R^*/k^* \leq (n+1)d - 1 = 3d - 1$*, so this bound is tight.*

Consider the relation

$$-x^d = y^d - 1 = \prod_{i=0}^{d-1}(y - \zeta_d^i)$$

which holds in R*, where* ζ_d *is a primitive* d*-th root of unity. From Lemma 2.23, we conclude that* $(y - \zeta_d^i)$ *is a unit in* R*, for all* $0 \leq i \leq d - 1$*. Interpreting the above relation as a dependency among* $x, y - \zeta_d, \ldots, y - \zeta_d^{d-1}$ *in* R^*/k^**, we can write any* $y - \zeta_d^i$ *multiplicatively in terms of* x *and* $y - \zeta_d^j$ *for* $j \neq i$*. Thus we can choose – for instance – to treat* $y - \zeta_d^{d-1}$ *as redundant, and we obtain new units* $y - \zeta_d^i$ *for* $0 \leq i \leq d - 2$*. Note that the relation above does not give a way to express* x *in terms of* $y - \zeta_d^i$*, since* x *appears with multiplicity* d*.*

In an analogous way, we may also rearrange the defining equation of R *to obtain*

$$-y^d = x^d - 1 = \prod_{i=0}^{d-1}(x - \zeta_d^i)$$

which gives new units $x - \zeta_d^i$ *for* $0 \leq i \leq d - 2$*. Finally, the rearrangement*

$$1 = x^d + y^d = \prod_{i=0}^{d-1}(x - \zeta_{2d}^{2i+1} y)$$

gives new units $x - \zeta_{2d}^{2i+1} y$ *for* $0 \leq i \leq d - 2$*.*

We thus have the units $x - \zeta_d^i$*,* $y - \zeta_d^i$*,* $x - \zeta_{2d}^{2i+1}$ *where* $0 \leq i \leq d - 2$*. In addition to the two units* x, y*, this gives a total of* $3(d - 1) + 2 = 3d - 1$ *units. Note that although we have accounted for obvious redundancies by removing* $x - \zeta_d^{d-1}, y - \zeta_d^{d-1}$*, and* $x + \zeta_{2d}^{2d-1} y$*, we have not yet shown that these* $3d - 1$ *units are independent. Algebraically, this would entail showing that there are no nontrivial multiplicative relations between these* $3d - 1$ *elements, a fairly nontrivial task. We instead adopt a geometric approach, whose utility will become evident already in this case.*

First, the divisors of these units (viewed as rational functions) are supported on the boundary ∂C *of the Fermat curve, which consists of the following* $3d$ *points:*

1. $P_i \coloneqq [\zeta_{2d}^{2i+1} \colon 1 \colon 0]$ *for* $0 \leq i \leq d - 1$

2. $Q_i \coloneqq [\zeta_d^i \colon 0 \colon 1]$ *for* $0 \leq i \leq d - 1$

3. $T_i := [0\colon \zeta_d^i\colon 1]$ *for* $0 \leq i \leq d-1$

As before, let $\phi : R^*/k^* \to \mathrm{Div}^0_\partial(\overline{X})$ *be the injection in* (2.3.1). *We have*

1. $\phi(x) = \sum T_i - \sum P_i$

2. $\phi(y) = \sum Q_i - \sum P_i$

3. $\phi(y - \zeta_d^j) = dT_j - \sum P_i$ *for* $0 \leq j \leq d-2$

4. $\phi(x - \zeta_d^j) = dQ_j - \sum P_i$ *for* $0 \leq j \leq d-2$

5. $\phi(x - \zeta_{2d}^{2j+1} y) = (d-1)P_j - \sum_{i \neq j} P_i$ *for* $0 \leq j \leq d-2$

Under the identification $\mathrm{Div}_\partial(\overline{C}) = \mathbb{Z}\langle P_1, \ldots, P_d, Q_1, \ldots, Q_d, T_1, \ldots, T_d\rangle \cong \mathbb{Z}^{3d}$*, we obtain the* $3d \times (3d-1)$ *matrix in Figure 2.1 whose columns represent the divisors of our given units.*

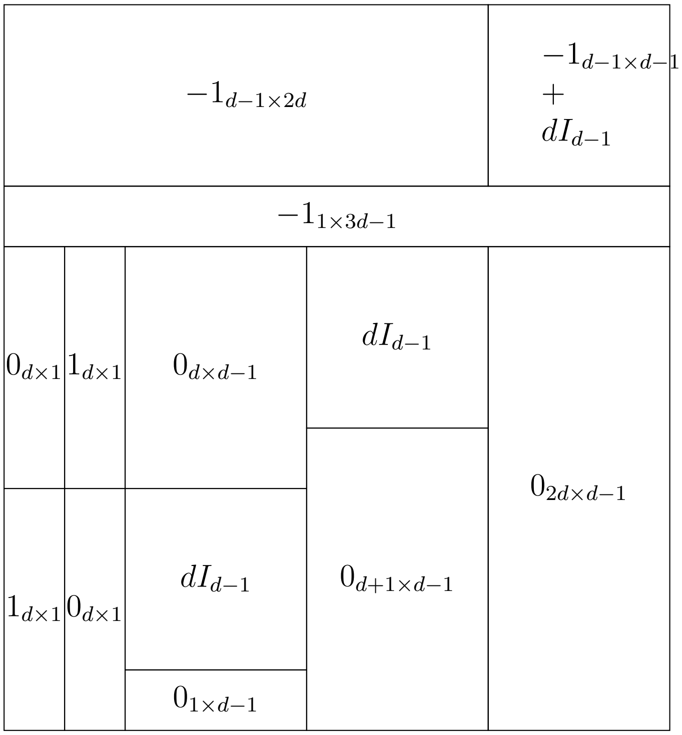


Figure 2.1: The block matrix whose columns are divisors of the units described in Example 2.24 for the Fermat curve $x^d + y^d = z^d$. Here $a_{m\times n}$ is an $m \times n$ matrix whose elements are all a, and I_n is the $n \times n$ identity matrix.

A straightforward check shows that this matrix has full rank $3d-1$, and therefore our units have no relations. It is natural at this point to ask whether these units form a basis for the unit group. It turns out that this need not be the case, as shown in Example 2.28.

Remark 2.25. *We observe several things about this computation. First, we did not necessarily compute generators of R^*/k^*. Instead, we found enough mutually independent elements to confirm a rank statement on R^*/k^*. Next, this technique was only effective for the Fermat curve because of special features of its defining equation. With more variables or nearly any perturbation of the defining equation, the method of obtaining units above fails. Finally, the argument above can only prove lower bounds on the rank of the unit group. We want to compute generators of the unit group, so in general we will need more tools than Lemma 2.23.*

Plane conics

Let $\overline{C} \subseteq \mathbb{P}^2_k$ be a smooth projective plane conic defined by a homogeneous quadric $f(x,y,z)$, and C the corresponding very affine curve (obtained by dehomogenizing with respect to z and intersecting with the 2-torus $\mathbb{T}^2 := \mathbb{A}^2 \setminus V(xy)$), with coordinate ring R. We describe methods for answering Questions 2.14 and 2.15 in this case. Combined with Lemma 2.22, this gives an algorithm to compute a basis of R^*/k^*. (Note that we can easily generate rational points on a plane conic.)

Algorithm 2.26 (Computing unit groups of conics).

```
Require: A homogeneous quadric $f(x,y,z)$ defining a plane conic $\overline{C} \subseteq \mathbb{P}^2$
Ensure: A basis of $R^*/k^*$
$P_1, \dots, P_n \leftarrow$ boundary points of $\overline{C}$
$P \leftarrow$ any other point of $\overline{C}$
for all $i \in \{1, \dots, n\}$ do
    $L_i \leftarrow$ defining equation of line between $P_i$ and $P$
end for
for all $i \in \{1, \dots, n-1\}$ do
    Compute $f_i \in k[x^{\pm 1}, y^{\pm 1}]$ equivalent to $L_i/L_{i+1}$ in $R$ using Algorithm 2.21
end for
return $f_1, \dots, f_{n-1}$
```

Theorem 2.27. *Algorithm 2.26 computes a basis for R^*/k^*.*

Proof. Observe that $\mathrm{Cl}^0(\overline{C}) = 0$ (as $\overline{C} \cong \mathbb{P}^1$). The exact sequence (2.3.1) then implies that the injection $R^*/k^* \hookrightarrow \mathrm{Div}^0_\partial(\overline{C})$ is an isomorphism. Then, note that $P_1 - P_2, \cdots, P_{n-1} - P_n$ forms a basis for $\mathrm{Div}^0_\partial(\overline{C})$, and L_i/L_{i+1} corresponds to the divisor $P_i - P_{i+1}$. Applying Algorithm 2.21 finishes the proof. □

Note that the choice of basis $\{P_i - P_{i+1}\}$ in the above proof was arbitrary; any basis of $\mathrm{Div}^0_\partial(\overline{C})$ would suffice. On the other hand, this basis gives the very simple rational functions L_i/L_{i+1}.

Example 2.28. *Consider the degree 2 Fermat curve $\overline{C}$ defined by $x^2 + y^2 = z^2$. We show that the units produced in Example 2.24 are not generators of R^*/k^*. As in Example 2.24, we have the following boundary points:*

1. $P_0 := [i\colon 1\colon 0]$
2. $P_1 := [-i\colon 1\colon 0]$
3. $Q_0 := [1\colon 0\colon 1]$
4. $Q_1 := [-1\colon 0\colon 1]$
5. $T_0 := [0\colon 1\colon 1]$
6. $T_1 := [0\colon -1\colon 1]$

Example 2.24 gives the following units and divisors (with $R^*/k^* \overset{\phi}{\hookrightarrow} \mathrm{Div}^0_\partial(\overline{C})$ as in (2.3.1)*):*

1. $\phi(x) = T_0 + T_1 - P_0 - P_1$
2. $\phi(y) = Q_0 + Q_1 - P_0 - P_1$
3. $\phi(y - 1) = 2T_0 - P_0 - P_1$
4. $\phi(x - 1) = 2Q_0 - P_0 - P_1$
5. $\phi(x - iy) = P_0 - P_1$

The subgroup of $\mathrm{Div}^0_\partial(\overline{C})$ generated by these divisors is given by the integer column span of the matrix, which is exactly Figure 2.1 for $d = 2$:

$$\begin{bmatrix} -1 & -1 & -1 & -1 & 1 \\ -1 & -1 & -1 & -1 & -1 \\ 0 & 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

As noted in the proof of Theorem 2.27, one basis for $\mathrm{Div}^0_\partial(\overline{C})$ *is* $\{P_i - P_{i+1} \mid 1 \leq i \leq n-1\} = P_1 - P_2, P_2 - P_3, \ldots, P_{n-1} - P_n$. *From this basis we obtain the matrix*

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

The first lattice has index 4 in the second. It follows that the units given in Example 2.24 are not generators in this case.

Example 2.29. *Let* $\overline{C}$ *be the conic defined by* $f = (1+t)x^2 + (1+t)y^2 + (1+t)z^2 - (2+2t+t^2)xy - (2+2t+t^2)yz - (2+2t+t^2)xz$, *where* k *is the field of Puiseux series in* t *over* $\mathbb{C}$. *Consider the very affine curve* C *given by intersecting with the canonical torus. Its boundary points are*

1. $P_1 := [0 : 1 : t+1]$
2. $P_2 := [0 : t+1 : 1]$
3. $P_3 := [1 : 0 : t+1]$
4. $P_4 := [t+1 : 0 : 1]$
5. $P_5 := [1 : t+1 : 0]$
6. $P_6 := [t+1 : 1 : 0]$

As described above, we can take a basis of $\mathrm{Div}^0_\partial(\overline{C})$ *to be differences of these boundary points, e.g.* $P_3 - P_1$, $P_3 - P_2$, $P_5 - P_3$, $P_5 - P_4$, *and* $P_6 - P_1$. *Algorithm 2.26 gives the following particularly nice generators of the unit group:*

1. $P_3 - P_1$ *gives* $f_1 :=$ *(line between* P_3 *and* P_2*)/(line between* P_1 *and* P_2*)* $= \frac{(t+1)^2x+y-(t+1)}{x} = (t+1)^2 + yx^{-1} - (t+1)x^{-1}$
2. $P_3 - P_2$ *gives* $f_2 :=$ *(line between* P_1 *and* P_3*)/(line between* P_1 *and* P_2*)* $= \frac{(t+1)x+(t+1)y-1}{x} = (t+1) + (t+1)yx^{-1} - x^{-1}$
3. $P_5 - P_3$ *gives* $f_3 :=$ *(line between* P_5 *and* P_4*)/(line between* P_3 *and* P_4*)* $= \frac{(t+1)x-y-(t+1)^2}{y} = (t+1)xy^{-1} - 1 - (t+1)^2y^{-1}$
4. $P_5 - P_4$ *gives* $f_4 :=$ *(line between* P_5 *and* P_3*)/(line between* P_3 *and* P_4*)* $= \frac{(t+1)x-y-1}{y} = (t+1)xy^{-1} - 1 - y^{-1}$

5. *$P_6 - P_1$ gives $f_5 :=$ (line between P_6 and P_2)/(line between P_1 and P_2) $= \frac{x-(t+1)y+(t+1)^2}{x} = 1-(t+1)yx^{-1}+(t+1)^2x^{-1}$*

So the intrinsic torus has dimension 5, and these generators specify a map into the intrinsic torus, corresponding to the ring map $\varphi\colon k[x_1^{\pm 1}, \dots, x_5^{\pm 1}] \to k[x^{\pm 1}, y^{\pm 1}]/\langle f \rangle$ sending $x_i \mapsto f_i$.

We note that the tropicalization of f is simply the tropical line $0 \oplus x \oplus y$ shown in the left in Figure 2.2.

We used Singular [20, 49] to compute the tropicalization of f in its intrinsic torus with basis equal to $\{x, y, f_1, f_2, f_3\}$. The intrinsic tropicalization has the following snowflake structure typical of a generic tropical conic as shown in the right in Figure 2.2.

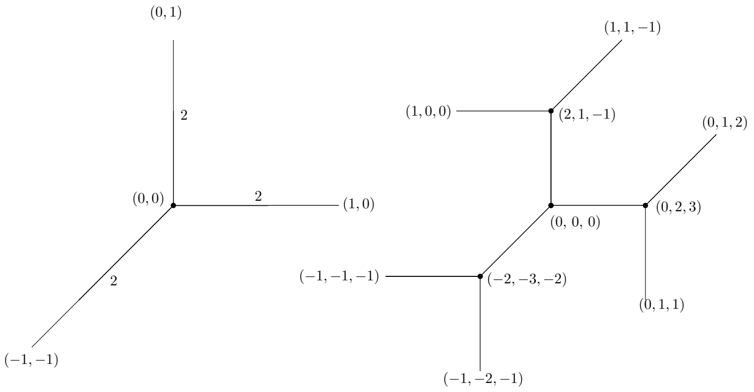


Figure 2.2: The tropicalization and intrinsic tropicalization respectively of the conic in Example 2.29, with coordinates and nontrivial multiplicities labeled.

Remark 2.30. *Consider the complete graph whose nodes are the elements of ∂C. Choose a spanning tree of this graph, and pick a direction for each edge. Each edge of this tree gives a divisor; namely an edge from P to Q gives the divisor $P - Q$. This gives a basis of* $\mathrm{Div}_\partial(\overline{C})$.

For instance, in Example 2.28, the basis

$$P_0 - Q_0, P_1 - Q_0, Q_1 - P_0, T_0 - P_0, T_1 - P_0$$

corresponds to the directed tree on the left in Figure 2.3.

Similarly, the basis

$$Q_0 - P_0, Q_0 - P_1, Q_0 - Q_1, Q_0 - T_0, Q_0 - T_1$$

corresponds to the tree on the right in Figure 2.3 (rooted at Q_0).

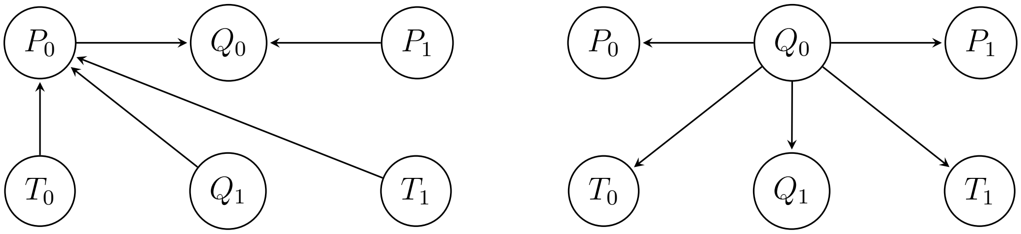


Figure 2.3: Two directed trees describing two different bases for the intrinsic torus of Example 2.28.

2.5 Rational normal curves

We next consider rational normal curves in parametric form. Recall that for any n, a rational normal curve $\overline{C}$ of degree n is the image of $\mathbb{P}^1$ under an embedding $\nu\colon \mathbb{P}^1 \hookrightarrow \mathbb{P}^n$ given by $\nu([S\colon T]) = [f_0(S,T) : \cdots : f_n(S,T)]$, where $f_0, \ldots, f_n$ are k-linearly independent homogeneous polynomials of degree n. Let $C := \overline{C} \cap \mathbb{T}^n$ be the corresponding very affine curve, with coordinate ring R. Our goal in this section is to give an algorithm for computing a basis of R^*/k^*.

Remark 2.31. *Plane conics are precisely the rational normal curves of degree 2, so the following discussion generalizes part of Section 2.4. Note though that the presentation of the curves in question has changed: here we do not begin with the implicit equations of the rational normal curve in $\mathbb{P}^n$.*

The following is a modification of the polynomial subalgebra membership algorithm given in [19, Proposition 7.3.7].

Algorithm 2.32 (Subalgebra membership).

```
Require: $f_0, \ldots, f_n$ degree $n$ homogeneous polynomials in $k[S,T]$ defining a rational normal curve, and a rational function $\frac{f}{g} \in k(S,T)$
Ensure: $\gamma \in k[x_1^{\pm 1}, \ldots, x_n^{\pm 1}]$ such that its homogenization $\overline{\gamma} \in k[x_0^{\pm 1}, \ldots, x_n^{\pm 1}]$ satisfies $\frac{f(S,T)}{g(S,T)} = \overline{\gamma}\big(f_0(S,T), \ldots, f_n(S,T)\big)$ if such a $\gamma$ exists, or false otherwise
$I \leftarrow \langle y_0 - f_0, \ldots, y_n - f_n, uf_0 \ldots f_n - 1, gs - 1\rangle$ in $k[y_0, \ldots, y_n, u, s, S, T]$
$G \leftarrow$ GröbnerBasis(I) in a monomial ordering where any monomial involving $s, S, T$ is greater than any monomial in $k[y_0, \ldots, y_n, u]$.
$h \leftarrow$ the remainder of dividing $fs$ by $G$
if $h \in k[y_0, \ldots, y_n, u]$ then
    return $h(1, x_1, \ldots, x_n, (x_1 \ldots x_n)^{-1})$
else
    return false
```

```
8: end if
```

Lemma 2.33. *Let $\overline{C}$ be a rational normal curve with parametrization $\psi\colon \mathbb{P}^1 \hookrightarrow \mathbb{P}^n$ given by $f_0, \dots, f_n$. Algorithm 2.32 correctly returns $\gamma \in k[x_1^{\pm 1}, \dots, x_n^{\pm 1}]$ such that its homogenization $\overline{\gamma} \in k[x_0^{\pm 1}, \dots, x_n^{\pm 1}]$ satisfies $\frac{f(S,T)}{g(S,T)} = \overline{\gamma}\big(f_0(S,T), \dots, f_n(S,T)\big)$ if such a γ exists.*

Proof. Note that it suffices to find the pushforward γ of a rational function $\frac{f(S,T)}{g(S,T)}$ on $\psi^{-1}(C)$ along the map given by $\psi^{-1}(C) \hookrightarrow C$, if such a γ exists and is regular. There exists $\overline{\gamma} \in k[x_0^{\pm 1}, \dots, x_n^{\pm 1}]$ such that

$$\frac{f}{g} = \overline{\gamma}\big(f_0, \dots, f_n\big)$$

if and only if there exists $\chi \in k[y_0, \dots, y_n, u]$ such that

$$\frac{f}{g} = \chi\big(f_0, \dots, f_n, (f_0 \dots f_n)^{-1}\big).$$

Setting $u = (f_0 \dots f_n)^{-1}$ and $s = g^{-1}$, this is equivalent to the statement that fs is in the k-algebra generated by $\{f_0, \dots, f_n, u\}$ in the ring $k[u, s, S, T]/\langle uf_0 \dots f_n - 1, gs - 1\rangle$. By [19, Proposition 7.3.7(i)], the previous statement is true if and only if h, the remainder upon dividing fs by the Gröbner basis G, is in the polynomial ring $k[y_0, \dots, y_n, u]$.

Suppose γ exists, let $\overline{\gamma}$ be its homogenization, and fix the notation of the previous paragraph. The argument above shows that h is a polynomial in $y_0, \dots, y_n, u$. By [19, Proposition 7.3.7(ii)], $fs = h(f_0, \dots, f_n, (f_0 \dots f_n)^{-1})$ is an expression of fs as a polynomial in $f_0, \dots, f_n, (f_0 \dots f_n)^{-1}$, so we can write $\overline{\gamma} = h(x_0, \dots, x_n, (x_0 \dots x_n)^{-1})$. Dehomogenizing, we get $\gamma = h(1, x_1, \dots, x_n, (x_1 \dots x_n)^{-1})$ as the pushforward of f/g. Because h is a Laurent polynomial, γ is regular on C. □

Algorithm 2.34 (Computing unit groups of rational normal curves)**.**

```
Require: A rational normal curve $\overline{C}$ given parametrically by $f_0(T,S), \dots, f_n(T,S) \in k[S,T]$
    and a corresponding very affine curve given by setting $f_0 = 1$
Ensure: A $\mathbb{Z}$-basis of $R^*/k^*$
$D \leftarrow \emptyset$
$[a_1 \colon b_1], \dots, [a_m \colon b_m] \leftarrow$ preimages of $\partial C$ under the parametrization map $\mathbb{P}^1 \hookrightarrow \mathbb{P}^n$.
Choose any basis $B$ of $\mathrm{Div}^0_\partial(\overline{C})$
for all $\sigma \in B$ do
    Write $\sum_i c_i[a_{k_i} \colon b_{k_i}] - \sum_j d_j[a_{l_j} \colon b_{l_j}]$ as the preimage of $\sigma$
    $f \leftarrow \prod_i (b_{k_i}S - a_{k_i}T)^{c_i}$
    $g \leftarrow \prod_j (b_{l_j}S - a_{l_j}T)^{d_j}$
    $\overline{\gamma} \leftarrow$ Algorithm 2.32 $\left(f_0, \dots, f_n, \frac{f}{g}\right)$
    $\gamma \leftarrow \overline{\gamma}(1, x_1, \dots, x_n)$
    $D \leftarrow D \cup \{\gamma\}$
end for
```

```
12: return D
```

Theorem 2.1. *Let $\overline{C} \subseteq \mathbb{P}^n_k$ be a rational normal curve over an algebraically closed field k, given parametrically as the image of a map $\mathbb{P}^1_k \hookrightarrow \mathbb{P}^n_k$. Let $C := \overline{C} \cap \mathbb{T}^n$ be the corresponding very affine curve, with coordinate ring R. Then Algorithm 2.34 correctly computes a $\mathbb{Z}$-basis of R^*/k^*.*

Proof. Let C be parametrized by $f_0(S,T), \dots, f_n(S,T) \in k[S,T]$. As $\overline{C} \cong \mathbb{P}^1$, $\mathrm{Cl}^0_\partial(\overline{C}) = 0$, so the injection $R^*/k^* \hookrightarrow \mathrm{Div}^0_\partial(\overline{C})$ is an isomorphism. For each basis element $\sigma = \sum_i c_i[a_{k_i} \colon b_{k_i}] - \sum_j d_j[a_{l_j} \colon b_{l_j}]$, Algorithm 2.32 produces a rational function $\overline{\gamma}$ on $\overline{C}$ which has zeros of order c_i at $[f_0(a_{k_i}, b_{k_i}) : \cdots : f_n(a_{k_i}, b_{k_i})]$ and poles of order d_j at $[f_0(a_{l_j}, b_{l_j}) : \cdots : f_n(a_{l_j}, b_{l_j})]$. Dehomogenizing gives the unit in R^* corresponding to σ. □

Example 2.35. *Consider the degree 3 rational normal curve $\overline{C} \subseteq \mathbb{P}^3$ given by the parametrization*

$$[S^3 - 4ST^2 : S^2T - 9T^3 : (S-3T)T^2 : (S+3T)T^2]$$

We compute the following boundary points (note that e.g. P_1, P_5, and P_6 are obtained by solving for $[S:T]$ when the first coordinate function $S^3 - 4ST^2$ is equal to zero):

1. $P_1 = [0\colon 1]$
2. $P_2 = [1\colon 0]$
3. $P_3 = [3 : 1]$
4. $P_4 = [-3 : 1]$
5. $P_5 = [2 : 1]$
6. $P_6 = [-2 : 1]$

We choose the following basis of $\mathrm{Div}^0_\partial(\overline{C})$*:*

$$P_1 - 2P_2 - P_4 + P_5 + P_6, P_2 - P_3, P_3 - P_4, P_4 - P_5, P_5 - P_6$$

Choose coordinates x, y, z, w on $\mathbb{P}^3$. We run Algorithm 2.32 to obtain preimages under $\overline{\phi}$ of our basis of $\mathrm{Div}^0_\partial(\overline{C})$ in $\mathrm{Frac}(\overline{R})^*$. Their corresponding dehomogenizations with respect to w give a basis of R^*/k^*:

1. $x \rightsquigarrow x$
2. $y \rightsquigarrow y$
3. $z \rightsquigarrow z$
4. $\dfrac{x + 5y + \frac{45}{6}(w-z) + 10(w+z)}{x} \rightsquigarrow \dfrac{x + 5y + \frac{45}{6}(1-z) + 10(1+z)}{x}$

5. $\dfrac{x-4y+6(w-z)+4(w+z)}{x} \rightsquigarrow \dfrac{x-4y+6(1-z)+4(1+z)}{x}$

Remark 2.36. *Although we do not do so here, one could consider various generalizations of the results presented thus far. For example, one can essentially perform the same procedure with "pinched" rational curves, i.e. smooth rational curves of degree $> n$ in $\mathbb{P}^n$. However, once higher-dimensional varieties or curves with singularities are considered, the situation becomes more complicated; even computing the boundary is no longer a simple task.*

2.6 Elliptic curves

Fix $k = \overline{\mathbb{Q}}$, let $\overline{E} \subseteq \mathbb{P}^2_k$ be an elliptic curve with a given base point O, and set $E := \overline{E} \cap \mathbb{T}^2$. Due to (2.3.1), computing the image of R^*/k^* in $\mathrm{Div}^0_\partial(\overline{E})$ is equivalent to computing the relations between the closed points of $\partial E =: \{P_1, \ldots, P_n\}$ in $\mathrm{Cl}^0_\partial(\overline{E})$. As the group law on the elliptic curve coincides with the group law in the class group, it suffices to compute relations between the corresponding points on the elliptic curve, which can be done via canonical Néron–Tate heights.

The Canonical Néron–Tate Height Pairing

We briefly define canonical Néron–Tate heights, following the exposition from [79]. Speaking broadly, height functions measure the "arithmetic complexity" of points on abelian varieties. For any field F and variety X, let $X(F)$ denote the F-rational points of X.

Theorem 2.37 ([79, p. VIII.9.3]). *Let $\overline{E}$ be an elliptic curve defined over a number field. There exists a unique function $\hat{h}\colon \overline{E}(\overline{\mathbb{Q}}) \to \mathbb{R}$ called the* canonical Néron–Tate height. *It satisfies the following properties:*

1. *For all $P, Q \in \overline{E}(\overline{\mathbb{Q}})$, the parallelogram law holds, i.e.*

$$\hat{h}(P+Q) + \hat{h}(P-Q) = 2\hat{h}(P) + 2\hat{h}(Q).$$

2. *For all $P \in E(\overline{\mathbb{Q}})$ and $m \in \mathbb{Z}$,*

$$\hat{h}(mP) = m^2\hat{h}(P).$$

3. *$\hat{h}$ is an even function, and the pairing*

$$\langle \quad , \quad \rangle \colon E(\overline{\mathbb{Q}}) \times E(\overline{\mathbb{Q}}) \to \mathbb{R}$$

$$\langle P, Q\rangle = \hat{h}(P+Q) - \hat{h}(P) - \hat{h}(Q)$$

is bilinear. This is equivalent to saying that $\hat{h}$ is a quadratic form on $E(\overline{\mathbb{Q}})$. We call this the canonical Néron–Tate height pairing.

4. For all $P \in \overline{E}(\overline{\mathbb{Q}})$, one has $\hat{h}(P) \geq 0$, and $\hat{h}(P) = 0$ if and only if P is torsion.

For any number field K, we can obtain a bilinear form on $\overline{E}(K)$ by restricting the bilinear form on $\overline{E}(\overline{\mathbb{Q}})$ in Theorem 2.37(3). This can be extended to a bilinear form on the finite-dimensional real vector space $\overline{E}(K) \otimes \mathbb{R}$.

Proposition 2.38 ([79, p. VIII.9.6])**.** *The Néron–Tate height induces a positive definite inner product on $\overline{E}(K) \otimes \mathbb{R}$.*

One can compute heights on elliptic curves efficiently with [70, Algorithm 6.1].

Computing Generators of the Unit Group

We now detail algorithms to solve Questions 2.14 and 2.15 for elliptic curves. First we treat Question 2.14. In addition to the above theory on Néron–Tate heights, we will need the following theorem and subroutines.

Theorem 2.39 ([65], [88, Theorem 4])**.** *Suppose L is a sublattice in $\mathbb{Z}^n$ of rank m. Fix some topological vector space norm on $\mathbb{R}^n$. For all $1 \leq k \leq m$, let M_k denote the minimum size ball centered at the origin that contains k linearly independent vectors in L. Then there exists a basis $\{x_1, \ldots, x_n\}$ of L such that for all $1 \leq k \leq m$, $|x_k| \leq (\frac{3}{2})^{k-1} M_k$.*

Subroutine 2.40 (Computing the relations among some torsion points of an elliptic curve)**.**

```
Require: Torsion points $T_1, \ldots, T_r$ on an elliptic curve and torsion orders $m_1, \ldots, m_n$
Ensure: Generators for the lattice of relations among $T_1, \ldots, T_r$ in $\mathbb{Z}^r$
D ← ∅
for all $(n_1, \ldots, n_r)$ where $0 \leq n_i \leq m_i$ do
    if $n_1T_1 + \cdots + n_rT_r = 0$ then
        add $(n_1, \ldots, n_r)$ to $D$
    end if
end for
return D
```

Subroutine 2.40 correctly computes all relations among a set of torsion points, as it simply manually checks all possible relations.

Subroutine 2.41 (Computing the relations modulo torsion of some torsion points on an elliptic curve)**.**

```
Require: A set of nontorsion points $Q_1, \ldots, Q_n$ on an elliptic curve
Ensure: Generators in $\mathbb{Z}^n$ for the lattice of relations among the $Q_i$ in $\overline{E}(\overline{\mathbb{Q}})/\operatorname{tors}$
Compute the $n \times n$ matrix $A$ such that the $A_{i,j} \leftarrow \langle Q_i, Q_j \rangle = \hat{h}(Q_i + Q_j) - \hat{h}(Q_i) - \hat{h}(Q_j)$
return generators of $\ker A \cap \mathbb{Z}^n$
```

Lemma 2.42. *Subroutine 2.41 correctly computes the lattice of relations among the nontorsion points $Q_1, \dots, Q_n$ in $\overline{E}(\overline{\mathbb{Q}})/\operatorname{tors}$.*

Proof. Choose some number field K large enough such that $\{Q_1, \dots, Q_n\} \subseteq \overline{E}(K)$. Note that $\overline{E}(K)$ modulo torsion embeds into $\overline{E}(K) \otimes \mathbb{R}$. By Theorem 2.37 (3), A is the inner product matrix of a nondegenerate inner product, and thus $\ker A \cap \mathbb{Z}^n$ comprises the relations among the Q_i up to torsion. □

We are now ready to solve Question 2.14 for elliptic curves.

Algorithm 2.43 (Answering Question 1 for elliptic curves)**.**

```
Require: An elliptic curve over $\overline{\mathbb{Q}}$ with a nonempty finite set of distinguished points $S \subseteq \overline{E}(\overline{\mathbb{Q}})$ and a base point $O$
Ensure: A minimal generating set of $\ker(\mathrm{Div}_S^0(\overline{E}) \to \mathrm{Cl}^0(\overline{E}))$
Determine torsion points of $S$ by Theorem 2.37(4). Let $Q_1, \dots, Q_n$ refer to nontorsion points of $S$, and let $T_1, \dots, T_r$ refer to torsion points of $S$ with orders $m_1, \dots, m_r$ respectively.
$D \leftarrow \emptyset \subseteq \mathbb{Z}^{n+r}$
$G \leftarrow$ finite subgroup generated by $(T_1, \dots, T_r) \subseteq \overline{E}(\overline{\mathbb{Q}})$
$D_T \leftarrow$ relations between $T_1, \dots, T_r$ as given by Subroutine 2.40
$D \leftarrow \{(0, \dots, 0, n_1, \dots, n_r) \in \mathbb{Z}^{n+r} \mid (n_1, \dots, n_r) \in D_T\}$
$D_Q \leftarrow$ relations modulo torsion between $Q_1, \dots, Q_n$ as given by Subroutine 2.41
$\ell \leftarrow \mathrm{rank}(\mathrm{span}_{\mathbb{Z}}(D_Q))$
$\lambda \leftarrow 0$, $S_\lambda \leftarrow \emptyset \subset \mathbb{Z}^n$
while $\mathrm{rank}(\mathrm{span}_{\mathbb{Z}}(S_\lambda)) \neq \ell$ do
    $\lambda \leftarrow \lambda + 1$
    $S_\lambda \leftarrow \left\{(m_1, \dots, m_n) \in \mathrm{span}_{\mathbb{Z}}(D_Q) \,\middle|\, \sqrt{\sum m_i^2} \leq \lambda \quad \text{and} \quad m_1Q_1 + \cdots + m_nQ_n \in G\right\}$
end while
$\Lambda \leftarrow \left\{(m_1, \dots, m_n) \in \mathrm{span}_{\mathbb{Z}}(D_Q) \,\middle|\, \sqrt{\sum m_i^2} \leq (\frac{3}{2})^{\ell-1}\lambda \quad \text{and} \quad m_1Q_1 + \cdots + m_nQ_n \in G\right\}$
for $(m_1, \dots, m_n) \in \Lambda$ do
    Choose $(n_1, \dots, n_r)$ such that $m_1Q_1 + \cdots + m_nQ_n + n_1T_1 + \cdots + n_rT_r = 0$
    add $(m_1, ..., m_n, n_1, \dots, n_r)$ to $D$
end for
Let $\deg : \mathbb{Z}^{n+r} \to \mathbb{Z}$ be the map sending $(x_1, \dots, x_{n+r}) \mapsto \sum x_i$
return a minimal set of generators for $\ker\left(\deg|_{\mathrm{span}_{\mathbb{Z}}(D)}\right)$
```

Lemma 2.44. *For a distinguished set S of $\overline{\mathbb{Q}}$-points on the elliptic curve E, Algorithm 2.43 correctly computes a minimal generating set of the kernel of the map $\mathrm{Div}_S^0(\overline{E}) \to \mathrm{Cl}^0(\overline{E})$.*

Proof. We first prove that the algorithm terminates. Let ψ denote the map $\mathrm{Div}_S(\overline{E}) \to \mathrm{Cl}_S(\overline{E})$, and let ψ_0 denote the restriction $\mathrm{Div}_S^0(\overline{E}) \to \mathrm{Cl}_S^0(\overline{E})$. Identify

$\mathrm{Div}_S^0(\overline{E}) \cong \mathbb{Z}\langle Q_1, \dots, Q_n, T_1, \dots, T_r\rangle$ with $\mathbb{Z}^{n+r}$ using this ordering of elements in S. For any subset $M \subseteq \{1, \dots, n+r\}$, let π_M denote the projection onto those coordinates.

Note that $\pi_{\{1,\dots,n\}}(\ker\psi) \subseteq \mathrm{span}_{\mathbb{Z}}(D_Q)$. In fact $\pi_{\{1,\dots,n\}}(\ker\psi)$ has the same rank as $\mathrm{span}_{\mathbb{Z}}(D_Q)$; if $(m_1, \dots, m_n) \in \mathrm{span}_{\mathbb{Z}}(D_Q)$, then $m_1Q_1 + \cdots + m_nQ_n \in G$ and thus is torsion. It follows that there exists some $m \in \mathbb{Z}$ such that $mm_1Q_1 + \cdots + mm_nQ_n = 0$, so that $(mm_1, \dots, mm_n) \in \pi_{\{1,\dots,n\}}(\ker\psi)$. Hence there exists a λ large enough to exit the while loop, and the algorithm terminates.

We now show the correctness of the algorithm. We claim that $\pi_{\{1,\dots,n\}}(\ker\psi) = \mathrm{span}_{\mathbb{Z}}(\Lambda)$. Note by definition that $\Lambda \subseteq \pi_{\{1,\dots,n\}}(\ker\psi)$ so $\mathrm{span}_{\mathbb{Z}}(\Lambda) \subseteq \pi_{\{1,\dots,n\}}(\ker\psi)$. By Theorem 2.39, as S_λ contains at least ℓ linearly independent elements, Λ will contain a lattice basis of $\pi_{\{1,\dots,n\}}(\ker\psi)$. Thus $\pi_{\{1,\dots,n\}}(\ker\psi) = \mathrm{span}_{\mathbb{Z}}(S)$.

Next we show that $\mathrm{span}_{\mathbb{Z}}(D) = \ker\psi$. Clearly $\mathrm{span}_{\mathbb{Z}}(D) \subseteq \ker\psi$ by construction. Suppose $(m_1, \dots, m_n, n_1, \dots, n_r) \in \ker\psi$. Then $(m_1, \dots, m_n) \in \pi_{\{1,\dots,n\}}(\ker\psi) = \mathrm{span}_{\mathbb{Z}}(\Lambda)$, so there exist $n_1', \dots, n_r'$ such that $(m_1, \dots, m_n, n_1', \dots, n_r') \in \mathrm{span}_{\mathbb{Z}}(D) \subseteq \ker\psi$. Thus, $(0, \dots, 0, n_1 - n_1', \dots, n_r - n_r') \in \ker\psi$. However, $(0, \dots, 0, n_1 - n_1', \dots, n_r - n_r') \in \mathrm{span}_{\mathbb{Z}}(D)$ because of Subroutine 2.40, so

$$(m_1, \dots, m_n, n_1', \dots, n_r') + (0, \dots, 0, n_1 - n_1', \dots, n_r - n_r') = (m_1, \dots, m_n, n_1, \dots, n_r) \in \mathrm{span}_{\mathbb{Z}}(D).$$

To conclude, we note that $\ker\psi_0 = \ker\psi \cap L = \mathrm{span}_{\mathbb{Z}}(D) \cap L$. □

We now turn our attention to answering Question 2.15. The following is an explicit version of Miller's algorithm, specialized to genus 1 [67]. For the following algorithm, we use $+$ and $-$ to denote addition and subtraction in the group of Weil divisors (as formal sums of points, not divisor classes), and use $\oplus$ and i to denote addition and inversion in the group law on the elliptic curve. For a point P and $n \in \mathbb{Z}$, the notation nP means $P + \ldots + P$, not $P \oplus \cdots \oplus P$.

Algorithm 2.45 (Answering Question 2 for elliptic curves)**.**

```
Require: An elliptic curve $\overline{E}$ with basepoint $O$ and a divisor $D = \sum_P n_P P \in \mathrm{Div}_\partial^0(\overline{E})$
Ensure: Whether $D$ is in the image of $R^*/k^*$, and an element of $\mathrm{Frac}(\overline{R})^*$ mapping to $D$
    if it is
$f \leftarrow 1$
while $|D| \neq 0$ do
    if $\exists P, Q$ such that $n_P, n_Q > 0$ and $P \neq -Q$ then
        $D \leftarrow D - (P + Q + (i(P) \oplus Q) - 3O)$
        $f \leftarrow fL$ where $L$ is the line through $P$ and $Q$
    else if $\exists P, Q$ such that $n_P, n_Q > 0$ and $P = -Q$ then
        $D \leftarrow D - (P + Q - 2O)$
        $f \leftarrow fL$ where $L$ is the line through $P$ and $Q$
    else if $\exists P, Q$ such that $n_P, n_Q < 0$ and $P \neq -Q$ then
        $D \leftarrow D + (P + Q + (i(P) \oplus Q) - 3O)$
        $f \leftarrow f/L$ where $L$ is the line through $P$ and $Q$
```

```
12:    else if $\exists P, Q$ such that $n_P, n_Q < 0$ and $P = -Q$ then
13:      $D \leftarrow D + (P + Q - 2O)$
14:      $f \leftarrow f/L$ where $L$ is the line through $P$ and $Q$
15:    else if $D$ is of the form $mP - nQ + oO$ for $m, n \geq 0$ then
16:      if $m \geq 2$ and $P$ has order $3$ then
17:        $D \leftarrow D - (3P - 3O)$
18:        $f \leftarrow fL$ where $L$ is the tangent line at $P$
19:      else if $m \geq 2$ and $P$ does not have order $2$ then
20:        $D \leftarrow D - (2P + i(P \oplus P) - 3O)$
21:        $f \leftarrow fL$ where $L$ is the tangent line at $P$
22:      else if $m \geq 2$ and $P$ has order $2$ then
23:        $D \leftarrow D - (2P - 2O)$
24:        $f \leftarrow fL$ where $L$ is the tangent line at $P$
25:      else if $n \geq 2$ and $Q$ has order $3$ then
26:        $D \leftarrow D + (3Q - 3O)$
27:        $f \leftarrow f/L$ where $L$ is the tangent line at $Q$
28:      else if $n \geq 2$ and $Q$ does not have order $2$ then
29:        $D \leftarrow D + (2Q + i(Q \oplus Q) - 3O)$
30:        $f \leftarrow f/L$ where $L$ is the tangent line at $Q$
31:      else if $n \geq 2$ and $Q$ has order $2$ then
32:        $D \leftarrow D - (2Q - 2O)$
33:        $f \leftarrow f/L$ where $L$ is the tangent line at $Q$
34:      else if $m = 1$ and $n = 1$ then
35:        $D \leftarrow D + (Q + i(Q) - 2O)$
36:        $f \leftarrow f/L$ where $L$ is the line through $Q$ and $-Q$
37:      else if ($m = 1$ and $n = 0$) or ($m = 0$ and $n = 1$) then
38:        return this divisor is not in the image of $R^*/k^*$
39:      end if
40:    end if
41:  end while
42:  return $f$
```

Lemma 2.46. *Algorithm 2.45 correctly determines whether a divisor D is in the image of R^*/k^* and computes an element f of $\mathrm{Frac}(\overline{R})^*$ mapping to D if so.*

Proof. Let ϕ denote the map $R^*/k^* \hookrightarrow \mathrm{Div}_0(\overline{E})$. Note that the quantity $D - \phi(f)$ is a loop invariant. Note additionally that during every execution of the loop, exactly one of the conditionals is satisfied; if line 3, 6, 9, and 12 are not satisfied, then D must be of the form $mP - nQ + oO$ for $m, n \geq 0$. If $D = mP - nQ + oO$ then exactly one of line 16, 19, 22, 25, 28, 31, 34, or 37 must be satisfied. $|D|$ is strictly reduced during each iteration unless line 34 or line 37 are satisfied. Line 37 terminates the program. Line 34 cannot be satisfied in two consecutive loops. Thus the algorithm will terminate.

Assume $D \in \phi(R^*/k^*)$. If at some point in execution $|D| = 0$, then as D is degree 0, $D = 0$ and so $D = \phi(f)$. If, during the execution of the algorithm, line 37 is satisfied, then some point is linearly equivalent to the origin, which is a contradiction. Hence the algorithm outputs an element f with the desired property.

Now assume $D \notin \phi(R^*/k^*)$. Because $D - \phi(f)$ is a loop invariant, we will never have $|D| = 0$. Because the algorithm terminates, it must terminate at line 37, as desired. □

Algorithm 2.47 (Computing unit groups of elliptic curves).

```
Require: An elliptic curve over $\overline{\mathbb{Q}}$ with boundary points $\partial$ and a base point $O$
Ensure: A basis of $R^*/k^*$
$V \leftarrow$ a generating set of the image of $R^*/k^* \hookrightarrow \mathrm{Div}^0_\partial(\overline{E})$ by Algorithm 2.43
$B \leftarrow \emptyset$
for all $v \in V$ do
    $f/g \leftarrow$ a rational function with divisor $v$ using Algorithm 2.45
    $B \leftarrow B \cup \{h\}$, where $h$ is a Laurent polynomial with the same divisor as $f/g$ by Algorithm 2.21
end for
return $B$
```

Theorem 2.2. *Let $k = \overline{\mathbb{Q}}$, let $\overline{E} \subseteq \mathbb{P}^2_k$ be an elliptic curve, and let $E := \overline{E} \cap \mathbb{T}^2$ be the corresponding very affine elliptic curve with coordinate ring R. Then Algorithm 2.47 correctly computes a $\mathbb{Z}$-basis of R^*/k^*.*

Proof. The correctness of Algorithm 2.47 follows by Lemmas 2.44, 2.46, and 2.22. □

Remark 2.48. *Many of the algorithms presented in this section are most easily implemented (e.g. in Sage [77]) for elliptic curves in Weierstrass form. Given a projective isomorphism of $\overline{E}$ to a Weierstrass form $\overline{W}$ as $\varphi\colon \overline{E} \to \overline{W}$, we can compute relations among the points in $\varphi(\partial E)$ using Algorithm 2.43. These relations can be pulled back by φ^{-1} to all relations among the points in ∂E, because φ induces an isomorphism $\mathrm{Div}^0_\partial(\overline{E}) \cong \mathrm{Div}^0_S(\overline{W})$.*

Example 2.49. *Let E be the very affine elliptic curve $E = \mathrm{Spec}(\overline{\mathbb{Q}_5}[x^{\pm 1}, y^{\pm 1}]/\langle y^2 - (x-1)(x+1)(x-4)\rangle)$ with basepoint $[0:1:0]$. We compute the following six boundary points of $\overline{E} \subseteq \mathbb{P}^2_{\overline{\mathbb{Q}_5}}$:*

1. $Q_1 \coloneqq [0:2:1]$
2. $Q_2 \coloneqq [0:-2:1]$
3. $T_1 \coloneqq [0:1:0]$
4. $T_2 \coloneqq [1:0:1]$
5. $T_3 \coloneqq [-1:0:1]$

6. $T_4 := [4 : 0 : 1]$

T_1 is the identity on E and has torsion order 1; T_2, T_3, and T_4 have torsion order 2; and Q_1 and Q_2 are nontorsion. From (2.3.1)*, we only need to find relations between the* 6 *points listed above. Note that E is defined over $\mathbb{Q}$, $E(\mathbb{Q}) \hookrightarrow E(\overline{\mathbb{Q}_5})$, and the points Q_i, T_i are defined over $\mathbb{Q}$. Thus, it is sufficient to compute relations in $E(\mathbb{Q})$, and so our previous results to compute relations still apply. Algorithm 2.47 yields the following generating set for the lattice of relations, with corresponding units:*

1. $[1, 1, 1, 1, -2, -2] \rightsquigarrow -y/x^2$

2. $[0, 2, 0, 0, -1, -1] \rightsquigarrow (x - 1)/x$

3. $[0, 0, 2, 0, -1, -1] \rightsquigarrow (x + 1)/x$

4. $[0, 0, 0, 2, -1, -1] \rightsquigarrow (x - 4)/x$

Note that it is easy to find these units by inspection, but to check that these form a basis of R^*/k^*, we rely on the algorithms given in this section. We can compute the tropicalization of the elliptic curve $E \subseteq \mathbb{T}^2$ defined by the equation $y^2 = (x - 1)(x + 1)(x - 4)$ to be three rays emerging from the origin, as pictured on the left in Figure 2.4.

Using the unit group basis $\{x, y, x - 1, x + 1\}$, we compute the intrinsic tropicalization of E in $\mathbb{T}^4$ with Singular, shown on the right in Figure 2.4:

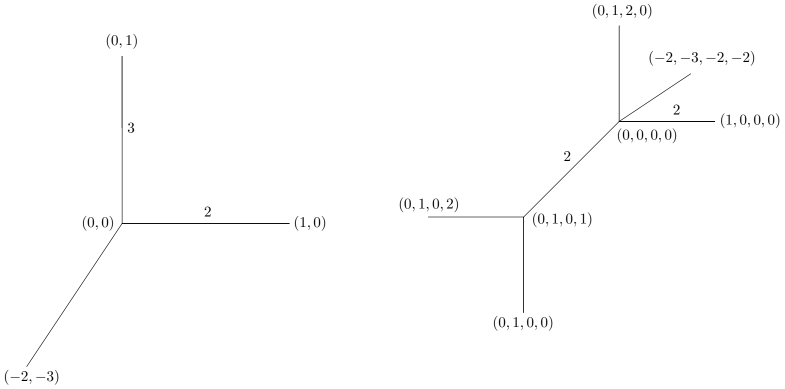


Figure 2.4: The tropicalization and intrinsic tropicalization respectively of the elliptic curve in Example 2.49, with coordinates and nontrivial multiplicities labeled.

In particular, the intrinsic tropicalization is larger than the original. On the other hand, the j-invariant is $j(E) =$ 438976/225, so that its* 5*-adic valuation is* −2*. It follows from Chan and Sturmfels [15] that $\overline{E}$ can be projectively re-embedded so that its tropicalization* *is in honeycomb form. The intrinsic tropicalization of E does not retain this information, as this projective re-embedding of $\overline{E}$ does not preserve our dehomogenization procedure. In particular, intrinsic tropicalizations need not be faithful.*

Hyperelliptic curves

The Néron–Tate canonical height can more generally be defined on any abelian variety defined over any number field. Fix some curve X defined over $\overline{\mathbb{Q}}$. Choose a number field K such that $\partial X \subseteq X(K)$. Letting J denote the Jacobian of X, the Néron–Tate canonical height pairing on $J(\overline{\mathbb{Q}})$ induces a positive definite inner product on $J(K) \otimes \mathbb{R}$. For curves of genus 2, Cassels, Flynn, and Smart provide an algorithm to compute the canonical height in [13] and [31], which has since been implemented in Magma [8]. For hyperelliptic curves of genus 3, Stoll [81] describes such an algorithm with a corresponding Magma implementation. Additionally, Holmes [47] has provided a height algorithm for all hyperelliptic curves. Another algorithm to compute heights for all hyperelliptic curves has been provided by Müller [69]. However, a hyperelliptic curve of the form $y^2 = f(x)$ in $\mathbb{P}^2$ with $\deg f \geq 4$ has a singularity at infinity, and thus the methods used for elliptic curves do not immediately generalize.

Conclusion

In this chapter, we developed algorithms for computing the unit groups for several classes of curves. We did so by embedding the unit group into the Weil divisor class group of the projective closure and analyzing the cokernel of this embedding as a subgroup of the divisor class group. Extending these methods to other classes of curves or to varieties of higher dimension will be a challenging topic of future research. Our motivation for calculating these unit groups was to compute the intrinsic tropicalization of very affine varieties; in the next section we focus on another algorithmic problem related to the tropicalization of varieties.

Chapter 3

Zero-dimensional tropical varieties via projections

This chapter is based on the manuscript "Computing zero-dimensional tropical varieties via projections" [37], which is joint work with Paul Görlach and Yue Ren, and which has been submitted to Computational Complexity.

3.1 Introduction

Computing tropical varieties of polynomial ideals is a fundamentally important yet algorithmically challenging task, requiring sophisticated techniques from computational algebra and convex geometry. Currently, Gfan [50] and Singular [20] are the only two programs capable of computing general tropical varieties. Both programs rely on a traversal of the Gröbner complex as initially suggested by Bogart, Jensen, Speyer, Sturmfels, and Thomas [7], and for both programs the initial bottleneck had been the computation of tropical links. Experiments suggest that this bottleneck was resolved with the recent development of new algorithms [14, 46]. However the new approaches still rely on computations that are known to be very hard, [14] on elimination and [46] on root approximation to an unknown precision.

In this chapter, we study the computation of zero-dimensional tropical varieties, which is the key computational ingredient in [46]. The computation uses triangular decomposition, which was also used in [46], and skew projections, which is the key conceptual idea behind [14]. The triangular decomposition splits the ideal into parts on which transformations can be efficiently applied. We show that the algorithm requires a polynomial number of field operations if we start with a Gröbner basis. In particular, we argue that in the computation of general tropical varieties, the calculation of tropical links becomes computationally insignificant compared to the Gröbner walk required to traverse the tropical variety.

Note that projections are a well-studied approach in polynomial systems solving, see [23, 82] for an overview on various techniques. Our approach can be regarded as a non-

Archimedean analogue of that strategy, since tropical varieties can be regarded as zeroth-order approximation of the solutions in the topology induced by the valuation.

This chapter is organized as follows: In Section 3.3, we introduce a special class of monomial transformations and study how they act on triangular sets. In Section 3.4, we explain our main algorithm for reconstructing zero-dimensional tropical varieties of triangular sets from their projections, while Section 3.5 analyzes the complexity of our algorithm. Section 3.7 touches upon some technical details of the implementation for the special case when the ideal is in shape position, and Section 3.8 compares the performance of our algorithm against a Magma implementation using univariate factorization and backsubstitution.

Implementations of all our algorithms can be found in the Singular library `tropicalProjection.lib`. Together with the data for the timings, it is available at https://mathrepo.mis.mpg.de/tropicalProjections/, and will also be made publicly available as part of the official Singular distribution.

3.2 Background

Let K be a field with non-trivial valuation $\nu\colon K^* \to \mathbb{R}$ and fix a multivariate polynomial ring $K[\mathbf{x}] := K[x_1,\dots,x_n]$ as well as a multivariate Laurent polynomial ring $K[\mathbf{x}^\pm] := K[x_1^\pm,\dots,x_n^\pm]$. Moreover, given a Laurent polynomial ideal $I \subseteq K[\mathbf{x}^\pm]$, we call a finite subset $G \subseteq I$ a *Gröbner basis* with respect to a monomial ordering $\prec$ on $K[\mathbf{x}]$ if G consists of polynomials and forms a Gröbner basis of the polynomial ideal $I \cap K[\mathbf{x}]$ with respect to $\prec$ in the conventional sense, see for example [39, §1.6]. Finally, a *lexicographical Gröbner basis* will be a Gröbner basis with respect to the lexicographical ordering $\prec_{\text{lex}}$ with $x_n \prec_{\text{lex}} \cdots \prec_{\text{lex}} x_1$.

For the sake of notation, we briefly recall some basic notions of computational algebra that are of immediate relevance to us. Our approach for computing zero-dimensional tropical varieties of multivariate ideals is based on computing sufficiently many projections to the univariate case.

For our approach, we will require the notion of triangular decomposition, a common concept for decomposing ideal into easier parts. Given a Gröbner basis, its time complexity is polynomial in the number of variables and the degree of the ideal [60]. In practice, it is fast enough to be a standard tool in some polynomial solvers such as in Maple [4].

Definition 3.1. *A* triangular set $T \subseteq K[\mathbf{x}]$ *is a finite set of polynomials, say* $T = \{g_n,\dots,g_1\}$*, where each* $g_i \in K[x_i,\dots,x_n]$ *is of the form* $g_i = x_i^{d_i} - f_i$ *for some* $d_i \in \mathbb{N}_{>0}$ *and* $f_i \in K[x_i,\dots,x_n]$ *with* x_j*-degree less than* d_j *for* $j \geq i$*. Note that this makes any triangular set a reduced lexicographical Gröbner basis.*

Proposition 3.2 ([39, Corollary 4.7.4])**.** *Let* $J \subseteq K[\mathbf{x}]$ *be a zero-dimensional polynomial ideal. Then there are triangular sets* $T_1,\dots,T_k \subseteq K[\mathbf{x}]$ *such that*

$$\sqrt{J} = \bigcap_{i\in[k]} \sqrt{\langle T_i\rangle} \quad \textit{and} \quad \langle T_i\rangle + \langle T_j\rangle = K[\mathbf{x}] \textit{ for } i \neq j.$$

Since each zero-dimensional ideal can be efficiently decomposed into triangular sets, we will focus on ideals that are generated by triangular sets. Moreover, we will put special emphasis on ideals in shape position.

Definition 3.3. *A zero-dimensional ideal $I \subseteq K[\mathbf{x}^{\pm}]$ is in* shape position *if it is generated by a triangular set $T = \{g_n, \dots, g_1\}$ with $d_i = 1$ for $i < n$, i.e., $g_i = x_i - f_i$ for a univariate polynomial $f_i \in K[x_n]$ for $i < n$.*

3.3 Unitriangular transformations on triangular sets

In this section, we consider special transformations on $K[\mathbf{x}^{\pm}]$ which arise from unitriangular transformations on the lattice of Laurent monomials, and describe how they operate on triangular sets. In Section 3.4, we will use these transformations to project a zero-dimensional tropical variety onto various lines.

Definition 3.4. *For any $u = (u_2, \dots, u_n) \in \mathbb{Z}^{n-1}_{\geq 0}$, we define a ring automorphism*

$$\varphi_u \colon K[\mathbf{x}^{\pm}] \to K[\mathbf{x}^{\pm}], \qquad x_i \mapsto \begin{cases} x_1 \cdot x_2^{-u_2} \cdots x_n^{-u_n} & \text{if } i = 1, \\ x_i & \text{if } i \neq 1, \end{cases}$$

and a linear projection

$$\pi_u \colon \mathbb{R}^n \twoheadrightarrow \mathbb{R}, \qquad (w_1, \dots, w_n) \mapsto w_1 + \sum_{i=2}^{n} u_i w_i.$$

We call such a φ_u a slim (unitriangular) transformation*.*

The reason we restrict ourselves to these simple transformations is because they allow us to compute a wide range of projections while being easy to use.

Lemma 3.5. *Let φ_u be a slim transformation. Then*

$$\pi_u(\operatorname{trop}(I)) = \operatorname{trop}(\varphi_u(I) \cap K[x_\ell^{\pm}]).$$

Proof. We may assume that K is algebraically closed. The ring automorphism φ_u induces a torus automorphism $f_u \colon (K^*)^n \xrightarrow{\sim} (K^*)^n$ with $f_u^{-1}(V(I)) = V(\varphi_u(I))$, which in turn induces a linear transformation $h_u \colon \mathbb{R}^n \xrightarrow{\sim} \mathbb{R}^n$ mapping $\operatorname{trop}(\varphi_u(I))$ to $\operatorname{trop}(I)$:

$$\begin{tikzcd}
x_1\prod_{i>1}x_i^{-u_i} & x_1 \arrow[l, mapsto] \\
K[\mathbf{x}^{\pm}] & K[\mathbf{x}^{\pm}] \arrow[l, "\varphi_u"']
\end{tikzcd}$$

induces

$$\begin{tikzcd}
(z_1,\dots,z_n) \arrow[r, mapsto] & (z_1\cdot\prod_{i>1}z_i^{-u_i},z_2,\dots,z_n) \\
(K^*)^n \arrow[r, "f_u"'] \arrow[d, "\nu"'] & (K^*)^n \arrow[d, "\nu"] \\
\mathbb{R}^n \arrow[r, "h_u"] & \mathbb{R}^n \\
(w_1,\dots,w_n) \arrow[r, mapsto] & (w_1-\sum_{i>1}u_iw_i,w_2,\dots,w_n)
\end{tikzcd}$$

Hence, with $p_1\colon\ \mathbb{R}^n\twoheadrightarrow\mathbb{R}$ denoting the projection onto the first coordinate:

$$\operatorname{trop}(\varphi_u(I)\cap K[x_1^{\pm}])=p_1(\operatorname{trop}(\varphi_u(I)))=(p_1\circ h_u^{-1})(\operatorname{trop}(I))=\pi_u(\operatorname{trop}(I)).\qquad\square$$

Algorithm 3.6 (Unitriangular transformations of triangular sets)**.**
Require: *(T,u), where*

- *T is a triangular generating set of a zero-dimensional ideal $I\subseteq K[\mathbf{x}^{\pm}]$,*
- *φ_u is a slim transformation.*

Ensure: *T', a triangular set generating $\varphi_u(I)$.*

```
Suppose $g_1 := x_1^{d_1}-\sum_{i=1}^{d_1}p_ix_1^{d_1-i}\in T$ with $p_i\in K[x_2,\dots,x_n]$.
for $i=1,\dots,d_1$ do
    $\hat{p}_i :=$ reduce$\big((x_2^{u_2}\cdots x_n^{u_n})^ip_i, T\setminus\{g_1\}\big)$.
end for
return $T' := T\setminus\{g_1\}\cup\{x_1^{d_1}-\sum_{i=1}^{d_1}\hat{p}_ix_1^{d_1-i}\}$.
```

Correctness of Algorithm 3.6. Only the last element in the triangular set

$$T=\Big\{x_n^{d_n}-f_n,\dots,x_2^{d_2}-f_2,\underbrace{x_1^{d_1}-\sum_{i=1}^{d_1}p_ix_1^{d_1-i}}_{=g_1}\Big\}$$

depends on x_1. Therefore, by replacing it with $\varphi_u(g_1)=(q^{-1}x_1)^{d_1}-\sum_{i=1}^{d_1}p_i(q^{-1}x_1)^{d_1-i}$, where $q:=x_2^{u_2}\dots x_n^{u_n}\in K[\mathbf{x}]$, we get generators of the transformed ideal $\varphi_u(I)\subseteq K[\mathbf{x}^{\pm}]$. Note that multiplying this element with the monomial q^{d_1} (which is invertible in $K[\mathbf{x}^{\pm}]$), we pass to the polynomial generating set of $\varphi_u(I)$ given by $T\setminus\{g_1\}\cup\{x_1^{d_1}-\sum_{i=1}^{d_1}p_iq^ix_1^{d_1-i}\}$. This is a non-reduced Gröbner basis with respect to the lexicographical ordering $\prec_{\mathrm{lex}}$, and replacing p_iq^i by $\hat{p}_i$, we reduce it modulo those generators not depending on x_1 to obtain the triangular set T'. $\square$

One special case that we would like to highlight separately is when an ideal I is in *shape position.* For ideals in shape position, Algorithm 3.6 simplifies drastically and performs quite well in practice, visible in the timings in Section 3.8. However, the complexity in Section 3.5 remains unchanged.

Remark 3.7 (Unitriangular transformations of ideals in shape position)**.** *If I is in shape position, it is generated by a triangular set T of the following form for $f_n, \dots, f_1 \in K[x_n]$:*

$$T = \{x_n^d - f_n,\, x_{n-1} - f_{n-1},\, \dots,\, x_2 - f_2,\, x_1 - f_1\}.$$

This has two main implications:

1. *It simplifies Algorithm 3.6. The triangular set generating $\varphi_u(I)$ will be of the form $T' = \{x_n^d - f_n,\, \dots,\, x_2 - f_2,\, x_1 - f_1'\}$, where $f_1' \in K[x_n]$ is the univariate polynomial with $\deg(f_1') < d$ and*

$$f_1' \equiv \Big(x_n^{-u_n} \cdot \prod_{i=2}^{n-1} f_i^{-u_i}\Big)^{-1} \cdot f_1 \equiv \Big(x_n^{u_n} \cdot \prod_{i=2}^{n-1} f_i^{u_i}\Big) \cdot f_1 \pmod{x_n^d - f_n}. \tag{3.3.1}$$

   In particular, $\varphi_u(I)$ will be in shape position.

2. *It allows us to use a wider range of transformations beyond those considered in Definition 3.4. To be precise, replacing f_1 with f_ℓ in Equation* (3.3.1) *we may use any transformation of the form*

$$\varphi_u\colon K[\mathbf{x}^\pm] \to K[\mathbf{x}^\pm], \qquad x_i \mapsto \begin{cases} x_1^{-u_1} \cdots x_\ell^{1} \cdots x_n^{-u_n} & \text{if } i = \ell, \\ x_i & \text{if } i \neq \ell. \end{cases}$$

   with $u_i \in \mathbb{N}$.

3.4 Computing zero-dimensional tropical varieties via projections

In this section, we assemble our algorithm for computing $\operatorname{trop}(I)$ for a zero-dimensional ideal $I \subseteq K[\mathbf{x}^\pm]$ generated by a triangular set. This is done in two stages, see Figure 3.1: In the first stage, we project $\operatorname{trop}(I)$ onto all coordinate axes of $\mathbb{R}^n$. In the second stage, we iteratively glue the coordinate projections together by projecting $\operatorname{trop}(I)$ onto more lines.

For the sake of simplicity, all algorithms contain some elements of ambiguity to minimize the level of technical detail. To see how these ambiguities are resolved in the actual implementation, see Section 3.7.

The following algorithm merges several small projections into a single large projection. For clarity, given a finite subset $A \subseteq \{1, \dots, n\}$, we use $\mathbb{R}^A$ to denote the linear subspace of $\mathbb{R}^n$ spanned by the unit vectors indexed by A and p_A to denote the projection $\mathbb{R}^n \twoheadrightarrow \mathbb{R}^A$. For $w \in \mathbb{R}^n$ and $I \subseteq K[\mathbf{x}^\pm]$, we denote $w_A := p_A(w) \in \mathbb{R}^A$ and $\operatorname{trop}(I)_A := p_A(\operatorname{trop}(I)) \subseteq \mathbb{R}^A$.

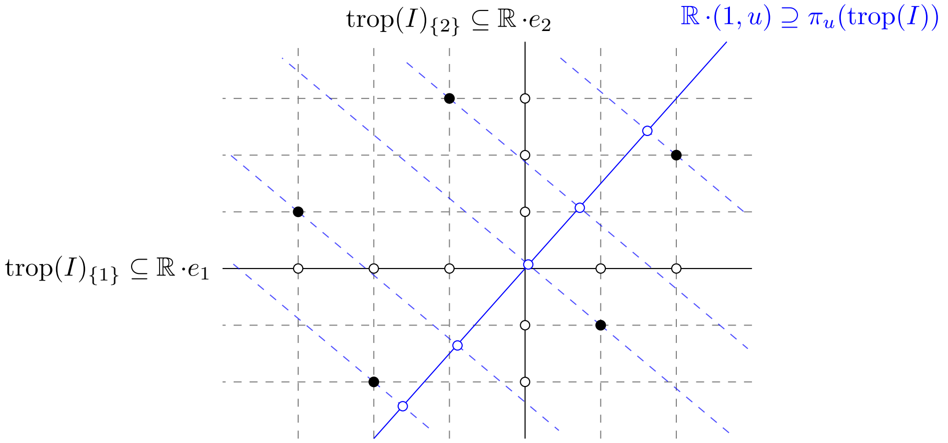

$\mathrm{trop}(I)_{\{2\}} \subseteq \mathbb{R}\cdot e_2$
$\mathbb{R}\cdot(1,u) \supseteq \pi_u(\mathrm{trop}(I))$
$\mathrm{trop}(I)_{\{1\}} \subseteq \mathbb{R}\cdot e_1$

Figure 3.1: Computing zero-dimensional tropical varieties via projections.

Algorithm 3.8 (gluing projections)**.**

Require: $(T, \mathrm{trop}(I)_{A_1}, \ldots, \mathrm{trop}(I)_{A_k})$, *where*

- *T is a triangular generating set of a zero-dimensional ideal $I \subseteq K[\mathbf{x}^\pm]$,*
- *$A_1, \ldots, A_k \subseteq \{1, \ldots, n\}$ are non-empty sets with $1 \in A := A_1 \cup \ldots \cup A_k$.*

Ensure: $\mathrm{trop}(I)_A \subseteq \mathbb{R}^A$.

```
1: Construct the candidate set
       $$C := \Big\{ w \in \mathbb{R}^A \;\Big|\; w_{A_i} \in \mathrm{trop}(I)_{A_i} \text{ for } i = 1, \ldots, k \Big\}.$$
2: Pick a $u \in \mathbb{Z}^{n-1}_{\geq 0}$ with $u_i = 0$ for $i \notin A$ such that the following map is injective:
       $$\pi_u|_C \colon C \to \mathbb{R}, \qquad (w_i)_{i\in A} \mapsto w_1 + \sum_{i\in A\setminus\{1\}} u_i w_i.$$
3: Using Algorithm 3.6, transform $T$ into a triangular set $T'$ generating $\varphi_u(I)$.
4: Compute the eliminant $\mu \in K[x_1]$, i.e., a generator of $\langle T'\rangle \cap K[x_1]$ and read off $\mathrm{trop}(\mu) \subseteq \mathbb{R}$ from its Newton polygon.
5: return $\{w \in C \mid \pi_u(w) \in \mathrm{trop}(\mu)\}$.
```

Correctness of Algorithm 3.8. First, we show the existence of a slim transformation φ_u required for Line 2 such that π_u is injective on the candidate set C. Extending the definition of the linear projection π_v from $v \in \mathbb{N}^{n-1}$ in Definition 3.4 to arbitrary $v \in \mathbb{R}^{n-1}$, it suffices to show that the set

$$Z := \{v \in \mathbb{R}^{n-1}_{\geq 0} \mid \pi_v|_T \text{ is injective}\} \subseteq \mathbb{R}^{n-1}$$

contains an integer point. By the definition of π_v, we see that

$$Z = \mathbb{R}^{n-1}_{\geq 0} \setminus \bigcup_{w \neq w' \in T} H_{w-w'}, \text{ where } H_{w-w'} := \Big\{ v \in \mathbb{R}^{n-1} \;\Big|\; \sum_{i=2}^{n} (w_i - w_i')v_i = w_1' - w_1 \Big\}.$$

This describes Z as the complement of an affine hyperplane arrangement in $\mathbb{R}^B$ inside the positive orthant. Therefore, Z must contain an integer point.

Next, note that the candidate set C contains $\mathrm{trop}(I)_A$ by construction, so the injectivity of $\pi_u|_T$ implies $\mathrm{trop}(I)_A = \{w \in T \mid \pi_u(w) \in \pi_u(\mathrm{trop}(I))\}$. Hence, the correctness of the output follows from $\pi_u(\mathrm{trop}(I)) = \mathrm{trop}(\mu)$, which holds by Lemma 3.5. □

The next algorithm computes $\mathrm{trop}(I)$ by projecting it onto all coordinate axes and gluing the projections together via Algorithm 3.8.

Algorithm 3.9 (tropical variety via projections)**.**

```
Require: $T$, a triangular generating set of a zero-dimensional ideal $I \subseteq K[\mathbf{x}^{\pm}]$.
Ensure: $\mathrm{trop}(I) \subseteq \mathbb{R}^n$
for $k \in \{1, \dots, n\}$ do
    Compute the eliminant $\mu_k \in K[x_k]$, i.e., a generator of $\langle T \rangle \cap K[x_k]$, and read off the projection $\mathrm{trop}(I)_{\{k\}} = \mathrm{trop}(\mu_k)$.
end for
Initialise a set of computed projections $W := \{\mathrm{trop}(I)_{\{1\}}, \dots, \mathrm{trop}(I)_{\{n\}}\}$.
while $W \not\ni \mathrm{trop}(I)_{\{1,\dots,n\}}$ do
    Pick projections $\mathrm{trop}(I)_{A_1}, \dots, \mathrm{trop}(I)_{A_k} \in W$ to be merged together such that $1 \in A$ and $\mathrm{trop}(I)_A \notin W$ for $A := A_1 \cup \dots \cup A_k$.
    Using Algorithm 3.8, compute $\mathrm{trop}(I)_A$.
    $W := W \cup \{\mathrm{trop}(I)_A\}$.
end while
return $\mathrm{trop}(I)_{\{1,\dots,n\}}$.
```

Correctness and Termination of Algorithm 3.9. In every iteration of the `while` loop, the set W grows in size. As there are only finitely many sets $A \subseteq \{1, \dots, n\}$, we will compute $\mathrm{trop}(I) = \mathrm{trop}(I)_{\{1,\dots,n\}}$ after finitely many iterations. □

Example 3.10. *Consider $K = \mathbb{Q}$ equipped with the 2-adic valuation and the ideal*

$$I = \langle \underbrace{2x_3^4 + x_3^3 + x_3^2 + x_3 + 2}_{=:g_3}, x_2 - \underbrace{2x_3}_{=:f_2}, x_1 - \underbrace{4x_3}_{=:f_1} \rangle \subseteq K[x_1^{\pm}, x_2^{\pm}, x_3^{\pm}].$$

This ideal is in shape position by Definition 3.3. From the Newton polygon of g_3, see Figure 3.2 (left), it is not hard to see that (for the sake of clarity, points with multiplicity 2 are *highlighted in bold):*

$$\begin{aligned}\operatorname{trop}(I)_{\{3\}} &= \operatorname{trop}(g_3) = \{-1, \mathbf{0}, 1\},\\ \operatorname{trop}(I)_{\{2\}} &= \{\lambda + 1 \mid \lambda \in \operatorname{trop}(I)_{\{3\}}\} = \{0, \mathbf{1}, 2\},\\ \operatorname{trop}(I)_{\{1\}} &= \{\lambda + 2 \mid \lambda \in \operatorname{trop}(I)_{\{3\}}\} = \{1, \mathbf{2}, 3\}.\end{aligned}$$

To merge $\operatorname{trop}(I)_{\{1\}}$ *and* $\operatorname{trop}(I)_{\{2\}}$*, we consider the following projection that is injective on the candidate set* $C := \operatorname{trop}(I)_{\{1\}} \times \operatorname{trop}(I)_{\{2\}}$*:*

$$\pi_{(3,0)}: \quad C \longrightarrow \mathbb{R}, \quad (w_1, w_2) \longmapsto w_1 + 3w_2.$$

The corresponding slim transformation $\varphi_{(3,0)}$ *sends* x_1 *to* $x_1 x_2^{-3}$ *and hence* $\varphi_{(3,0)}(I)$ *is generated by* $\{g_3, x_2 - f_2, x_1 x_2^{-3} - 4x_3\}$*, which Algorithm 3.6 transforms into the following lexicographical Gröbner basis:*

$$\varphi_{(3,0)}(I) = \Big\langle g_3,\, x_2 - f_2,\, x_1 - \underbrace{(-16x_3^3 - 16x_3^2 - 16x_3 - 32)}_{=: f_1'} \Big\rangle.$$

The eliminant in $K[x_1]$ *of* $\varphi_{(3,0)}(I)$ *can be computed as the resultant*

$$\operatorname{Res}_{x_3}(g_3, x_1 - f_1') = 8x_1^4 + 752x_1^3 + 32256x_1^2 + 770048x_1 + 8388608.$$

Figure 3.2 (middle) shows the Newton polygon of the resultant, from which we see:

$$\operatorname{trop}(\operatorname{Res}_{x_3}(g_3, x_1 - f_1')) = \{9, \mathbf{5}, 1\}.$$

Thus,

$$\operatorname{trop}(I)_{\{1,2\}} = \{(3,2), (\mathbf{2}, \mathbf{1}), (1,0)\}.$$

To merge $\operatorname{trop}(I)_{\{1,2\}}$ *and* $\operatorname{trop}(I)_{\{3\}}$*, we consider the following projection that is injective on the candidate set* $C := \operatorname{trop}(I)_{\{1,2\}} \times \operatorname{trop}(I)_{\{3\}}$*:*

$$\pi_{(0,3)}: \quad C \longrightarrow \mathbb{R}, \quad (w_1, w_2, w_3) \longmapsto w_1 + 3w_3.$$

The corresponding slim transformation $\varphi_{(0,3)}$ *sends* x_1 *to* $x_1 x_3^{-3}$ *and hence* $\varphi_{(0,3)}(I)$ *is generated by* $\{g_3, x_2 - f_2, x_1 x_3^{-3} - 4x_3\}$*, which Algorithm 3.6 transforms into the following lexicographical Gröbner basis:*

$$\varphi_{(0,3)}(I) = \Big\langle g_3,\, x_2 - f_2,\, x_1 - \underbrace{(-2x_3^3 - 2x_3^2 - 2x_3 - 4)}_{=: f_1''} \Big\rangle.$$

Another resultant computation yields the eliminant in $K[x_1]$ *of* $\varphi_{(0,3)}(I)$*:*

$$\operatorname{Res}_{x_3}(g_3, x_1 - f_1'') = 8x_1^4 + 94x_1^3 + 504x_1^2 + 1504x_1 + 2048.$$

Figure 3.2 (right) shows the Newton polygon of the resultant, from which we see:

$$\operatorname{trop}(\operatorname{Res}_{x_3}(g_3, x_1 - f_1'')) = \{6, \mathbf{2}, -2\},$$

and thus

$$\operatorname{trop}(I) = \operatorname{trop}(I)_{\{1,2,3\}} = \{(3,2,1), (\mathbf{2}, \mathbf{1}, \mathbf{0}), (1,0,-1)\}.$$

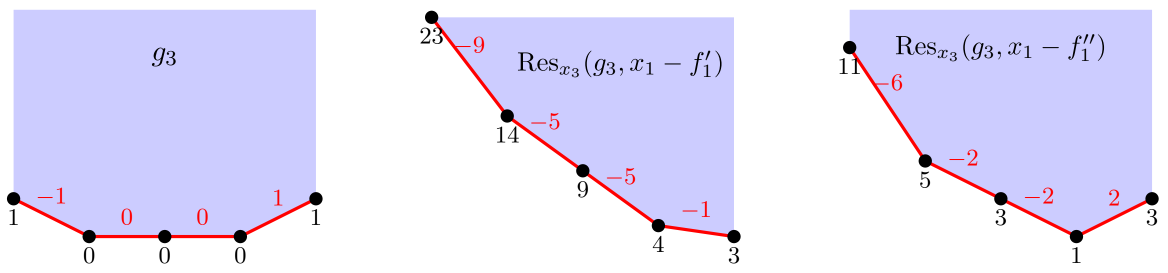

g_3
1
-1
0
0
1
1
0
0
0
$\text{Res}_{x_3}(g_3, x_1 - f_1')$
23
-9
14
-5
9
-5
4
-1
3
$\text{Res}_{x_3}(g_3, x_1 - f_1'')$
11
-6
5
-2
3
-2
1
2
3

Figure 3.2: Newton polygons of g_3 and the resultants in Example 3.10. Below each vertex is its height, above each edge is its slope.

Remark 3.11 (Eliminants for ideals in shape position)**.** *Note that, if an ideal I is in shape position, say generated by $\{x_n^d - f_n, \ldots, x_1 - f_1\}$ with $f_i \in K[x_n]$, then computing eliminants such as in Line 4 of Algorithm 3.8 and Line 2 of Algorithm 3.9 becomes much simpler. To compute the eliminant of I in $K[x_\ell]$, it suffices to consider the two polynomials $x_n^d - f_n, x_\ell - f_\ell \in K[x_\ell, x_n]$, making the computation independent of the number of variables n.*

3.5 Complexity

In this section, we bound the complexity for computing a zero-dimensional tropical variety $\text{trop}(I) \subseteq \mathbb{R}^n$ of an ideal generated by a given triangular generating set using Algorithm 3.9 with the `sequential` strategy. The `sequential` strategy sequentially computes the projections $\text{trop}(I)_{\{1\}}$, $\text{trop}(I)_{\{1,2\}}$, $\ldots$ until $\text{trop}_{\{1,\ldots,n\}} = \text{trop}(I)$. For more details on strategies, see Section 3.7. We show that the number of required arithmetic operations is polynomial in the degree of the ideal and the number of variables.

Combined with the `FGLM` algorithm [28] and Lazard's lextriangular decomposition [60], this shows that the tropical variety of any zero-dimensional ideal can be computed from its reduced Gröbner basis using polynomially many arithmetic operations, see Corollary 3.20.

Convention 3.12. *For the remainder of the section, we assume that $\nu(K^*) \subseteq \mathbb{Q}$, so that $\text{trop}(I) \subseteq \mathbb{Q}^n$.*

For the sake of convenience, we recall some well-known results on the complexity of arithmetic operations over integral extensions.

Proposition 3.13 ([35, Corollary 4.6 + Section 4.3])**.** *Let R be a ring and let $f \in R[z]$ be a monic univariate polynomial of degree d. Then:*

1. *Adding and multiplying in $R[z]/\langle f\rangle$ require at most $\mathcal{O}(d^2)$ arithmetic operations in R.*

2. *Computing the qth power in $R[z]/\langle f\rangle$ requires at most $\mathcal{O}(d^2 \log q)$ arithmetic operations in R.*

Corollary 3.14. *Given a triangular generating set $T \subseteq K[x_1,\dots,x_n]$ of a zero-dimensional ideal I of degree d:*

1. *Addition and multiplication in $K[x_1,\dots,x_n]/I$ require at most $\mathcal{O}(d^2)$ arithmetic operations in K.*
2. *Computing the q-th power in $K[x_1,\dots,x_n]/I$ requires at most $\mathcal{O}(d^2\log q)$ arithmetic operations in K.*

Proof. Suppose $T = \{g_n,\dots,g_1\}$ with $g_i \in K[x_i,\dots,x_n]$. Let $T_k := T \cap K[x_k,\dots,x_n]$, $R_k := K[x_k,\dots,x_n]/\langle T_k\rangle$ for $k = n,\dots,1$, and $T_{n+1} := \emptyset$, $R_{n+1} := K$. Note that T_k is a triangular set in R_k, and that R_k is an integral extension of R_{k+1} given by

$$R_k = R_{k+1}[x_k]/\langle g_k\rangle.$$

For $k \leq n$, Proposition 3.13 implies that addition and multiplication in R_k can be carried out using at most $\mathcal{O}(d_k^2)$ arithmetic operations in R_{k+1} or $\mathcal{O}(d_k^2\cdots d_n^2)$ arithmetic operations in K, and that computing a q-th power in R_{k+1} requires at most $\mathcal{O}(d_k^2\log q)$ arithmetic operations in R_{k+1} or $\mathcal{O}(d_k^2\cdots d_n^2\log q)$ arithmetic operations in K. The claimed bounds follow from $d = d_1\cdots d_n$. □

Proposition 3.15. *Given a triangular generating set $T \subseteq K[x_1,\dots,x_n]$ of a zero-dimensional ideal I of degree d and a slim transformation φ_u, Algorithm 3.6 computes the triangular generating set of $\varphi_u(I)$ using at most $\mathcal{O}\big(d^2\sum_{u_i>0}\log(u_i)\big)$ arithmetic operations in K.*

Proof. Consider the reductions in Algorithm 3.6 Line 3. Reducing $(x_2^{u_2}\dots x_n^{u_n})^i p_i$ by $T\setminus\{g_1\}$ is equivalent to expressing it in $K[x_2,\dots,x_n]/\langle T\setminus\{g_1\}\rangle$ as a linear combination of the K-basis $B := \{x_2^{b_2}\cdots x_n^{b_n} \mid 0 \leq b_i < d_i\}$.

By Corollary 3.14 and since $i \leq d_1$, expressing $(x_2^{u_2}\dots x_n^{u_n})^i$ in terms of B requires at most

$$\begin{aligned}\mathcal{O}\Big(\big(\sum_{u_i>0} d_i^2\cdots d_n^2\log(u_i)\big) + (|\{i \mid u_i > 0\}| - 1)\cdot d_2^2\cdots d_n^2 + d_2^2\cdots d_n^2\log(d_1)\Big)\\ \leq \mathcal{O}\Big(\log(d_1)d_2^2\cdots d_n^2\sum_{u_i>0}\log(u_i)\Big)\end{aligned}$$

operations in K. As $d^2 = d_1^2d_2^2\cdots d_n^2$, repeating the computation for $i = 1,\dots,d_1$ requires at most $\mathcal{O}\big(d^2\sum_{u_i>0}\log(u_i)\big)$ operations in K. The multiplications by p_i for $i = 1,\dots,d_1$ do not change the complexity. □

Lemma 3.16. *Given a triangular generating set $T \subseteq K[x_1,\dots,x_n]$ of a zero-dimensional ideal I of degree d, the computation of the eliminant $\mu \in K[x_k]$ of I, i.e., a generator of $I \cap K[x_k]$, requires at most $\mathcal{O}(d^3)$ arithmetic operations in K.*

Proof. Note that the eliminant $\mu \in K[x_k]$ is also the minimal polynomial of $x_k \in K[x_1,\dots,x_n]/I$. Hence it can be computed by finding a linear relation among the powers $1, x_k, x_k^2, \dots, x_k^{d-1}$. By Corollary 3.14, computing all powers requires $\mathcal{O}(d^3)$ arithmetic operations in K and, by [10, Chapter 16], computing a linear dependency requires $\mathcal{O}(d^{\omega+\varepsilon})$, where $\omega < 3$ is the exponent of the complexity of matrix multiplication and $\varepsilon > 0$. □

Lemma 3.17. *Let $X, Y \subseteq \mathbb{Q}$ be finite sets of cardinality $\leq d$. Then there exists a non-negative integer $m \leq \binom{d^2}{2}$ such that $X \times Y \to \mathbb{Q}$, $(a,b) \mapsto a - mb$ is injective. The smallest such m can be found in $\mathcal{O}(d^4)$ arithmetic operations in $\mathbb{Q}$.*

Proof. The map $(a,b) \mapsto a - mb$ will fail to be injective if and only if there exists a pair of points in $X \times Y$ lying on an affine line with slope m. Since there are at most $\binom{d^2}{2}$ pairs of points, the statement follows by the pigeonhole principle.

We can determine all integral slopes attained by a line between any two points of $X \times Y$ with $\mathcal{O}(\binom{d^2}{2}) = \mathcal{O}(d^4)$ arithmetic operations in $\mathbb{Q}$. Picking the smallest natural number not occurring among these slopes gives the desired m. □

The following proposition deals with the k-th call of Algorithm 3.8 in Line 7 of Algorithm 3.9 running the `sequential` strategy. Recall that the strategy sequentially computes the projections $\operatorname{trop}(I)_{\{1\}}$, $\operatorname{trop}(I)_{\{1,2\}}$, … until $\operatorname{trop}_{\{1,\dots,n\}} = \operatorname{trop}(I)$.

Proposition 3.18. *Let I be any zero-dimensional ideal in $K[x_1,\dots,x_n]$ of degree d. Let $k \in \{2,\dots,n\}$ and suppose Algorithm 3.8 is called with input*

- *T, a triangular generating set of a zero-dimensional ideal $I \subseteq K[\mathbf{x}^{\pm}]$,*
- $\operatorname{trop}(I)_{\{1,\dots,k-1\}}$ *and* $\operatorname{trop}(I)_{\{k\}}$.

Moreover, assume that the following are known from the previous call of Algorithm 3.8 in Line 7 of Algorithm 3.9 running the `sequential` strategy:

- *$\varphi_{u'}$, a slim transformation such that $\pi_{u'}$ is injective on* $\operatorname{trop}(I)_{\{1,\dots,k-1\}}$,
- *T', the triangular generating set of $\varphi_{u'}(I)$.*

Then Algorithm 3.8 for gluing the two projections into $\operatorname{trop}(I)_{\{1,\dots,k\}}$ *requires at most $\mathcal{O}(d^2(n+d))$ and $\mathcal{O}(d^4)$ arithmetic operations in K and $\mathbb{Q}$, respectively.*

Proof. Applying Lemma 3.17 to $X := \pi_{u'}(\operatorname{trop}(I)_{\{1,\dots,k-1\}})$ and $Y := \operatorname{trop}(I)_{\{k\}}$, we can compute a minimal $m \leq \binom{d^2}{2}$ such that $(a,b) \mapsto a - mb$ is injective on $X \times Y$ in $\mathcal{O}(d^4)$ arithmetic operations in $\mathbb{Q}$. Setting $u := u' + me_k$, this means that π_u is injective on $\operatorname{trop}(I)_{\{1,\dots,k-1\}} \times \operatorname{trop}(I)_{\{k\}}$.

Since $\varphi_u(I) = \varphi_v(\varphi_{u'}(I))$ for $v := me_k - e_\ell$ and the triangular generating set of $\varphi_{u'}(I)$ is already known, we may compute the triangular generating set T' of $\varphi_w(I)$ by applying Algorithm 3.6 to the input T' and φ_v. By Proposition 3.15, this requires $\mathcal{O}(nd^2 \log m) = \mathcal{O}(nd^2 \log d)$ arithmetic operations in K.

By Lemma 3.16, computing the eliminant $\mu \in K[x_1]$ requires $\mathcal{O}(d^3)$ arithmetic operations in K, so the overall number of arithmetic operations in K required for Algorithm 3.8 is $\mathcal{O}(d^2(n+d))$. □

Theorem 3.19. *Let I be any zero-dimensional ideal in $K[x_1,\ldots,x_n]$ of degree d. Algorithm 3.9, which computes the zero-dimensional tropical variety* $\mathrm{trop}(I)$*, with the* ***`sequential`*** *strategy requires at most $\mathcal{O}(nd^2(n+d))$ and $\mathcal{O}(nd^4)$ arithmetic operations in K and $\mathbb{Q}$, respectively.*

Proof. Running Algorithm 3.9 with the `sequential` strategy consists of the following nontrivial operations:

- computing eliminants $\mu_k \in K[x_k]$ for $k = 1,\ldots,n$ in Line 2,
- applying Algorithm 3.8 to $\mathrm{trop}(I)_{\{1,\ldots,k-1\}}$ and $\mathrm{trop}(I)_{\{k\}}$ for $k = 2,\ldots,n$ in Line 6.

Combining Lemma 3.16 and Proposition 3.18 then yields the stated bounds. □

Corollary 3.20. *Let I be any zero-dimensional ideal in $K[x_1,\ldots,x_n]$ of degree d. Given any Gröbner basis of I, computing* $\mathrm{trop}(I)$ *requires at most polynomially many (in terms of n and d) arithmetic operations in K and $\mathbb{Q}$ respectively.*

Proof. Using polynomially many arithmetic operations in K, any Gröbner basis may be transformed into a lexicographical Gröbner basis by [28], and any lexicographical Gröbner basis may be decomposed into triangular sets by [60]. The claim then follows from Theorem 3.19. □

Remark 3.21 (Comparison with Magma)**.** *In the Section 3.8, we compare timings of Algorithm 3.9 to the* Magma *script in Section 3.6 in the special case that the ideal is in shape position, i.e., generated by a set*

$$\{x_n^d - f_n,\ x_{n-1} - f_{n-1},\ \ldots,\ x_1 - f_1\} \quad \textit{for some } f_n,\ldots,f_1 \in K[x_n].$$

Our algorithm was implemented with the practical optimizations outlined in Remark 3.7 and Remark 3.11. The Magma *script uses a p-adic approximation of the roots of the univariate polynomial $x_n^d - f_n$ and substitution into $f_{n-1},\ldots,f_1$.*

It is difficult to compare the two implementations in terms of complexity due to their fundamentally different nature. As the Magma *script factorizes $x_n^d - f_n$, its complexity depends on the valuation of the discriminant f_n [33]. Moreover, the root approximations needs to be of sufficiently high precision to determine the valuation of the substituted polynomials.*

In the best case, such as when the generating set is a tropical basis [46, Proposition 2.16], the Magma *script terminates instantaneously:*

- *the valuations of the roots of $x_n^d - f_n$ are distinct and thus may be read off the slopes of its Newton polygon,*

- *the valuations of $f_{n-1}(z), \dots, f_1(z)$, z a root of f_n, are uniquely determined by the valuation of z.*

In the worst case, the valuation of the discriminant of $x_n^d - f_n$ is exponential in d and so is the complexity for its factorization.

3.6 Magma comparison

The comparison of timings in Section 3.8 is based on the following Magma implementation by Avi Kulkarni for finite precision p-adic fields. The function assumes that the ideal is in shape position and uses univariate factorization and substitution.

```
function pAdicSolutionsOverSplittingField(I, Qp)
  R := Generic(I);
  gs := GroebnerBasis(I);     //assumed to be in shape position

  u := UnivariatePolynomial(gs[#gs]);
  up := ChangeRing(u,Qp);
  K := SplittingField(up);    //main bottleneck of the algorithm

  vars_padic := Variables(ChangeRing(R,K));
  padic_rts := Roots(ChangeRing(up,K));

  function backSolve(rt)
    rt_coords := [rt];
    for i in [#gs-1 .. 1 by -1]  do
      g := Evaluate(gs[i], vars_padic[1..i] cat rt_coords);
      rti := Roots(UnivariatePolynomial(g));
      assert #rti eq 1;
      Insert(~rt_coords, 1, rti[1][1]);
    end for;
    return rt_coords;
  end function;

  return [ backSolve(rt[1]) : rt in padic_rts], K;
end function;
```

3.7 Implementation

In this section, we reflect on some design decisions that were made in the implementation of the algorithms in the Singular library `tropicalProjection.lib`. The library contains an implementations of Algorithms 3.6, 3.8, and 3.9 for the special case that the ideal is in shape position, as discussed in Remarks 3.7 and 3.11. While the reader who is only interested in the algorithms and their complexity may skip this section without impeding their understanding, we thought it important to include this section for the reader who is interested in the actual implementation.

Picking slim transformations in Algorithm 3.8 Line 2

As $\pi_u|_T$ is injective for generic $u \in \mathbb{Z}^{n-1}_{\geq 0}$, it seems reasonable to sample random $u \in \mathbb{Z}^{n-1}_{\geq 0}$ until the corresponding projection is injective on the candidate set. Our implementation however iterates over all $u \in \mathbb{Z}^{n-1}_{\geq 0}$ in increasing ℓ_1-norm until the smallest one with injective $\pi_u|_T$ is found. This is made in an effort to keep the slim transformation $\varphi_u(I)$ as simple as possible, since Lines 3–4 are the main bottlenecks of our algorithm.

Transforming Gröbner bases in Algorithm 3.8 Line 3

As mentioned, Lines 3–4 are the main bottlenecks of our algorithm. Two common reasons why polynomial computations may scale badly are explosions in degree or in coefficient size.

Note that the degree of the polynomials is unproblematic in our algorithm: By Remark 3.7.(1), using Algorithm 3.6 in Line 3 only incurs basic arithmetic operations in $K[x_n]/\langle x_n^d - f_n\rangle$ whose elements can be represented by polynomials of degree less than d. Also, the degree of the eliminant in Line 4 is naturally bounded by d. Coefficient explosion can be a problem in large examples, which is why we choose u as small as possible for the transformation.

Computing eliminants in Algorithm 3.8 Line 4 and Algorithm 3.9 Line 2

The computation of eliminants of an ideal in shape position, say generated by $\{x_n^d - f_n, x_{n-1} - f_{n-1}, \ldots, x_1 - f_1\}$ with $f_i \in K[x_n]$, can be carried out in many different ways. For example:

Resultants We can compute the resultant of the two polynomials $x_n^d - f_n$ and $x_\ell - f_\ell \in K[x_\ell, x_n]$ with respect to the variable x_n by standard resultant algorithms. The eliminant $\mu_\ell \in K[x_\ell]$ lies somewhere between the resultant and its squarefree part. In particular, the tropical variety of the eliminant is the tropical variety of the resultant.

Minimal polynomial Note that the eliminant $\mu_\ell \in K[x_\ell]$ is also the minimal polynomial of $\overline{f}_\ell \in K[x_n]/\langle x_n^d - f_n\rangle$. Hence it can be computed by standard minimal polynomial algorithms.

Gröbner bases Note that $\{x_n^d - f_n, x_\ell - f_\ell\} \subseteq K[x_\ell, x_n]$ is a Gröbner basis with respect to the lexicographical ordering with $x_n \prec x_\ell$. We can transform this to a Gröbner basis with respect to the lexicographical ordering with $x_\ell \prec x_n$ and read off the eliminant μ from it.

For polynomials with small coefficients, the implementation using SINGULAR's resultants seemed the fastest, but SINGULAR's `FGLM` [28] seems to be best when dealing with very large coefficients, though that may be due to implementation.

For $K = \mathbb{Q}$ however, we can use a modular approach thanks to the SINGULAR library `modular.lib` [80]: It computes the eliminants over $\mathbb{F}_p$ for several primes p using any of the above methods, then lifts the results to $\mathbb{Q}$. This modular approach avoids problems caused by very large coefficients and works particularly well using the method based on minimal polynomials from above. We can check if the lifted μ_ℓ is correct by testing whether $\mu_\ell(\overline{f}_\ell) = 0$ in $K[x_n]/\langle x_n^d - f_n \rangle$.

Picking gluing strategies in Algorithm 3.9 Line 6

Algorithm 3.9 is formulated in a flexible way. Different strategies of realizing the choice of coordinate sets $A_1, \ldots, A_k$ in Line 6 can adapt to the needs of a specific tropicalization problem. The four gluing strategies that are implemented in our Singular library are (see Figure 3.3 for an illustration in the case $n = 5$):

`oneProjection` Only a single iteration of the `while` loop, in which we pick $k = n$ and $A_i = \{i\}$ for $i = 1, \ldots, n$.

`sequential` $n-1$ iterations of the `while` loop, during which we pick $k = 2$ and $A_1 = \{1, \ldots, i\}$ and $A_2 = \{i+1\}$ in the i-th iteration.

`regularTree(`k`)` $n-1$ iterations of the `while` loop, which can be partially run in parallel in $\lceil \log_k n \rceil$ batches. In each batch we merge k of the previous projections. Note that, by Remark 3.7 (2), the condition that $1 \in A_1 \cup \ldots A_k$ is unnecessary if the ideal is in shape position.

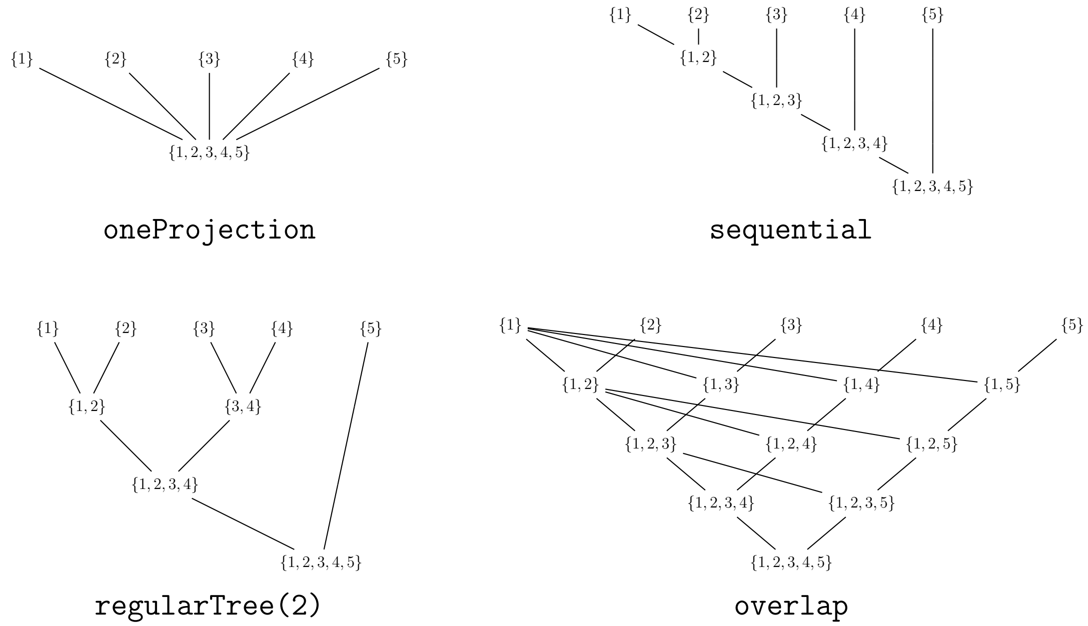
`overlap` $(n-1)n/2$ iterations of the `while` loop, which can be partially run in parallel in $n-1$ batches. During batch i, we pick $k = 2$ and $A_1 = \{1, \ldots, i\}$, $A_2 = \{1, \ldots, i-1, j\}$ for $j > i$.

`oneProjection` is the simplest strategy, requiring only a single slim transformation. For examples of very low degree, it is the best strategy due to its minimal overhead. For examples of higher degree d, the candidate set C in Algorithm 3.8 can become quite large, at worst $|C| = d^n$. This generally leads to larger $u \in \mathbb{Z}_{\geq 0}^{n-1}$ in Line 2 and causes problems due to coefficient growth.

`sequential` avoids the problem of a large candidate set C by only gluing two projections at a time, guaranteeing $|C| \leq d^2$. This comes at the expense of computing $n-1$ slim transformations, but even for medium-sized instances we observe considerable improvements compared to `oneProjection`. In Section 3.5, we have proved that `sequential` guarantees good complexity bounds on Algorithm 3.9.

`regularTree(`k`)` can achieve a considerable speed-up by parallelization. Whereas every `while`-iteration in `sequential` depends on the output of the previous iteration, the strategy `regularTree(`k`)` allows us to compute all gluings in parallel in $\lceil \log_k n \rceil$ batches. The total number of gluings remains the same.

`overlap` further reduces the size of the candidate set C compared to `sequential`, while exploiting parallel computation like `regularTree(`k`)`. It glues projections two at a time, but



`oneProjection` `sequential` `regularTree(2)` `overlap`

Figure 3.3: Visualisation of different gluing strategies.

only those A_1 and A_2 which overlap significantly. This can lead to much smaller candidate sets T, at best $|T| = d$, which makes a slim transformation obsolete. The strategy `overlap` seems particularly successful in practice and is the one used for the timings in Section 3.8.

Our implementation in Singular also allows for custom gluing strategies by means of specifying a graph as in Figure 3.3.

3.8 Timings

In this section we present timings of our Singular implementation of Algorithm 3.9 for $K = \mathbb{Q}$ and the 2-adic valuation. We compare it to a Magma [8] implementation by Avi Kulkarni which uses univariate factorization and backsubstitution, see Section 3.6. Note that Magma operates under finite absolute precision, which was chosen to be 2^{1000} by default and was increased if needed. While Singular is also capable of the same task, we chose to compare to Magma instead as it is significantly faster due to its finite precision arithmetic over p-adic numbers. Our Singular timings use the `overlap` strategy, a modular approach and parallelization with up to four threads. The Singular times we report on are total CPU times across all threads (for reference, the longest example in Singular required 118 seconds total CPU time, but only 32 seconds real time). All computations were run on a server with 2 Intel Xeon Gold 6144 CPUs, 384GB RAM and Debian GNU/Linux 9.9 OS. All examples and scripts are available at `https://mathrepo.mis.mpg.de/tropicalProjections`.

Random lexicographical Gröbner bases in shape position

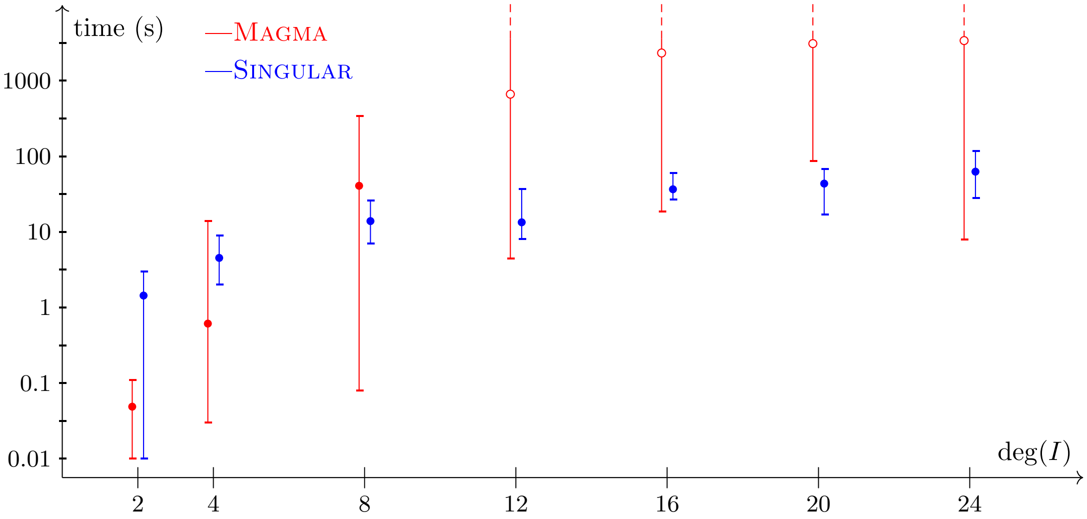
Given $d, n \in \mathbb{Z}_{>0}$, we construct random lexicographical Gröbner bases $G \subseteq \mathbb{Q}[x_1, \ldots, x_n]$ of degree d in shape position of the form

$$G = \{x_n^d - f_n,\, x_{n-1} - f_{n-1},\, x_{n-2} - f_{n-2},\, \ldots,\, x_2 - f_2,\, x_1 - f_1\},$$

where $f_n, \ldots, f_1$ are univariate polynomials in x_n of degree $d-1$ with coefficients of the form $2^\lambda \cdot (2k+1)$ for random $\lambda \in \{0, \ldots, 99\}$ and random $k \in \{0, \ldots, 4999\}$.

Figure 3.4 shows timings for $n = 5$ and varying d. Each computation was aborted if it failed to terminate within one hour. We see that Magma is significantly faster for small examples, while Singular scales better with increasing degree.

For many of the ideals I however, $\mathrm{trop}(I)$ has fewer than d distinct points. This puts our algorithm at an advantage, as it allows for easier projections in Algorithm 3.9 Line 2. Mathematically, it is not an easy task to generate non-trivial examples with distinct tropical points. Picking $x_n^d - f_n$ to have d roots with distinct valuation for example would make all roots live in $\mathbb{Q}_2$, in which case Magma terminates instantly. Our next special family of examples has criteria which guarantee distinct points.



deg(I)	2	4	8	12	16	20	24
#Singular finished	100	100	100	100	100	100	100
#Magma finished	100	100	100	93	51	21	9
Singular avg. (s)	1	5	14	19	37	44	63
Magma avg. (s)	0	1	41	>663	>2273	>3095	>3395

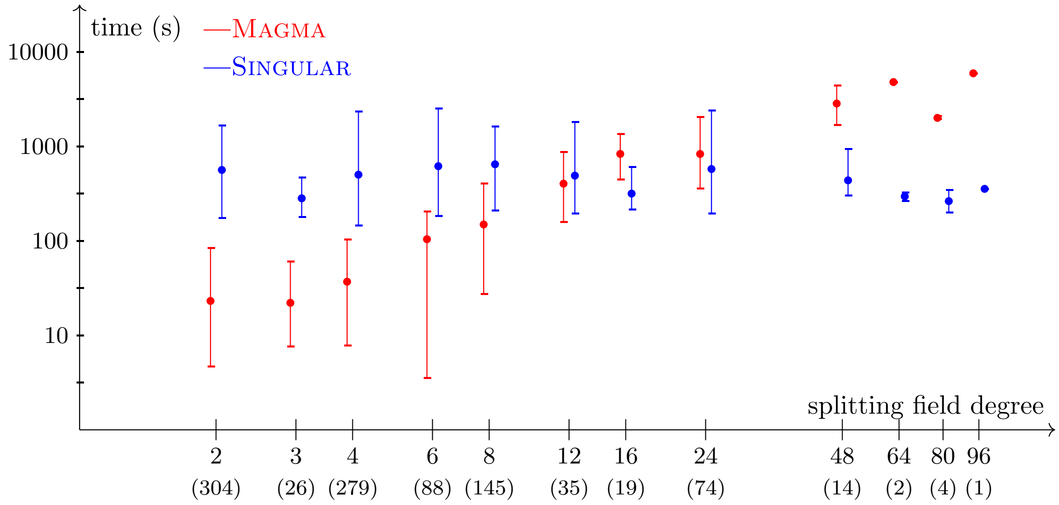
Figure 3.4: Timings for the randomly generated ideals in shape position.

Tropical lines on a random honeycomb cubic

Let $V(f) \subseteq \mathbb{P}^3$ be a smooth cubic surface. In [72], it is shown that $\operatorname{trop}(f) \subseteq \mathbb{R}^3$ may contain infinitely many tropical lines. However, for general f whose coefficient valuations induce a honeycomb subdivision of its Newton polytope, $\operatorname{trop}(f)$ will always contain exactly 27 distinct tropical lines [72, Theorem 27], including tropicalizations of the 27 lines on $V(f)$.

We used Polymake [36] to randomly generate 1000 cubic polynomials with honeycomb subdivisions whose coefficients are pure powers of 2. For each cubic polynomial f, we constructed the one-dimensional homogeneous ideal $\mathcal{L}_f \subseteq \mathbb{Q}[p_{12}, p_{13}, p_{14}, p_{23}, p_{24}, p_{34}]$ of degree 27 whose solutions are the lines on $V(f)$ in Plücker coordinates. Figure 3.5 shows the timings for computing $\operatorname{trop}(L_f)$, where $L_f := \mathcal{L}_f + \langle p_{34} - 1\rangle$ is a zero-dimensional ideal of degree 27. Out of our 1000 random cubics, 8 had to be discarded because L_f was of lower degree, i.e., $V(f)$ contained lines with $p_{34} = 0$.

Unsurprisingly, the Singular timings are relatively stable, while the Magma timings heavily depend on the degree of the splitting field of L_f over $\mathbb{Q}_2$. While the generic splitting field degree is 51840 over $\mathbb{Q}$ [26], the distinct tropical points of $\operatorname{trop}(L_f)$ severely restrict the Galois group of the splitting field over $\mathbb{Q}_2$.



splitting deg.	2	3	4	6	8	12	16	24	48	64	80	96
Frequency	304	26	279	88	145	35	19	74	14	2	4	1
Singular avg.	556	281	505	610	651	490	313	580	440	294	261	352
Magma avg.	23	22	37	104	149	403	831	830	2840	4791	1998	5935

Figure 3.5: Timings for the 27 tropical lines on a tropical honeycomb cubic.

3.9 Discussion

If a zero-dimensional ideal $I \subseteq K[x_1^\pm, \dots, x_n^\pm]$ is generated by a given triangular set, we have shown that the tropical variety $\operatorname{trop}(I) \subseteq \mathbb{R}^n$ can be computed using at most $\mathcal{O}(nd^2(n+d))$ and $\mathcal{O}(nd^4)$ arithmetic operations in K and $\mathbb{Q}$, respectively. Given a Gröbner basis of a general zero-dimensional ideal, it is thus possible to compute its tropical variety using polynomially many arithmetic operations.

For the special case that the ideal is in shape position, we have implemented our algorithms in Singular using parallelization and modular techniques, and we have compared its timings for $K = \mathbb{Q}_p$ to a Magma implementation using univariate factorization and backsubstitution. The timings of our algorithm are relatively constant, while the Magma timings depend strongly on the degree of the splitting field: For small degrees it severely outperformed our algorithm, while for large degrees it is significantly slower.

We would like to conclude the chapter with a remark on the complexity of computing tropical varieties $\operatorname{trop}(I)$ for $I \subseteq K[x_1, \dots, x_n]$ and $d := \deg(I) > 0$. Currently, there are two distinct methods for computing tropical varieties:

- reconstruction from sufficiently many projections of the tropical variety [44] ,
- traversing the Gröbner complex along the tropical variety [7, 66].

Not much is known about the complexity of either method. In fact not much is known about the combinatorial complexity of tropical varieties in general [54]. However, both methods generally involve Gröbner basis computations, though the projections in the first method may also be computed projection using numerical techniques if $K = \mathbb{C}\{\{t\}\}$ and the ideal is generated over $\mathbb{C}$ [9].

Currently, gfan and Singular are the only software systems capable of computing positive-dimensional tropical varieties, and both use the traversal of the Gröbner complex. The algorithm for the traversal method consists of two main parts:

- a Gröbner walk algorithm to walk from one Gröbner polyhedra to another,
- a tropical link algorithm to direct the Gröbner walk along the tropical variety.

While the computation of tropical links had been a major bottleneck of the original algorithm and in early implementations, experiments suggest that it has since been resolved by new approaches [14, 46]. However, the algorithm in [14, §4.2] relies heavily on projections, while [46, Algorithm 2.10] relies on root approximations to a possibly exponential precision, so neither approach has good complexity bounds.

Algorithm 3.9 was designed with [46, Algorithm 2.10] in mind. Replacing [46, Algorithm 2.10] in [46, Algorithm 4.6] with our Algorithm 3.9 allows us to compute tropical links at the cost of $2n$ Gröbner basis computations of zero-dimensional ideals of degree d, which is at most single exponential in the number of variables n [58, 59]. The Gröbner walk however requires Gröbner bases computations of initial ideals with respect to weight vectors

$w \in \operatorname{trop}(I)$ with $\dim C_w(I) = \dim \operatorname{trop}(I) - 1$, where $C_w(I)$ denotes the Gröbner polyhedron of I around w. These initial ideals are of the same dimension as I and neither monomial, as $w \in \operatorname{trop}(J)$, nor binomial, as $\dim C_w(J) < \dim \operatorname{trop}(J)$. Therefore, the Gröbner basis computations are double exponential in the number of variables n. Thus, the complexity of computing tropical varieties via a traversal of the Gröbner complex is dominated by the complexity of the necessary Gröbner walk, and the complexity of computing tropical links is irrelevant.

Conclusion

In this chapter, we described new algorithms for the calculation of zero-dimensional tropical varieties. Given an ideal I in triangular form, we described well-chosen unimodular transformations of the polynomial ring which allow us to easily calculate the projection of the variety onto one-dimensional lines. We showed that our algorithm requires polynomially many field operations if given a Gröbner basis and discussed implications for the complexity of computing general tropical varieties. The next two chapters of this dissertation describe applications of tropical geometry to other fields, demonstrating the importance of such algorithms for the computation of tropical objects.

Chapter 4

Min-convex hulls in the affine building

The original work in this chapter appeared in modified form as “Computing min-convex hulls in the affine building of SL_d”, which will be published in Discrete & Computational Geometry [91].

4.1 Introduction

Affine buildings are infinite simplicial complexes originally introduced by Bruhat and Tits to study the structure of simple Lie groups. They have since found use in a variety of other contexts, such as arithmetic geometry [56] and algebraic geometry [12].

We consider the affine building $\mathcal{B}_d$ associated to the group $SL_d(K)$ over a discrete valued field K. There is a natural notion of min-convex hull in $\mathcal{B}_d$, which provides a geometric data structure for the relations among $d \times d$ invertible matrices over K. Originally introduced by Faltings [27], this data structure underlies Mustafin varieties [12, 40] and can be used to study the fundamental group of certain 3-manifolds [83]. Joswig, Sturmfels, and Yu [51, Algorithm 2] give a procedure for computing such a min-convex hull in $\mathcal{B}_d$ as the standard triangulation of a tropical convex hull in some tropical projective space. However, their algorithm requires the enumeration of all lattice points in the min-convex hull under consideration, which can be difficult to implement and is expensive in practice. We devise an improved algorithm with time complexity bounded in the dimension of the building and the number of matrices spanning the min-convex hull, making it feasible for the first time to compute min-convex hulls in practice.

We briefly describe the structure of this chapter. In Section 4.2 we review the basics of min-convex lattice theory and tropical geometry that we rely on throughout. We review an algorithm for computing an apartment containing two vertices and develop its application to our problem in Section 4.4. We then describe our novel algorithm and prove its correctness in Section 4.5. In Section 4.6 we discuss an improvement on the previous algorithm when computing the min-convex hull of three lattice classes. Our algorithms have been implemented over the rational function field as a Polymake extension [36]. Algorithm 4.38

has also been implemented in Mathematica over the field of rational numbers with a p-adic valuation. This software and the code for the examples in this chapter can be found at our supplementary materials webpage:

`https://github.com/leonyz/min-convex-hull`

4.2 Background

We begin by fixing notation and reviewing the setup of [51]. Let K be a field with discrete valuation val : $K \to \mathbb{Z} \cup \{\infty\}$, R its valuation ring of elements with nonnegative valuation, π a uniformizer of valuation 1, and $k = R/(\pi)$ the residue field. For example, we might have $R = \mathbb{Z}_p$ be the p-adic integers for some prime number p. If so, then the p-adic numbers $K = \mathbb{Q}_p$ would be its field of fractions, $\pi = p$ is a uniformizer, and $k = \mathbb{Z}_p/(p) = \mathbb{F}_p$ is the residue field. Note that for any choice of field K, the vector space K^d is an R-module in a natural way.

Definition 4.1. *A* lattice Λ *is an R-submodule of K^d generated by d linearly independent vectors in K^d. We often represent a lattice by an invertible matrix whose columns generate the lattice.*

Let $\Lambda_1, \Lambda_2 \subseteq K^d$ be two lattices. We say that Λ_1 and Λ_2 are equivalent *if there exists $c \in K^*$ such that $\Lambda_1 = c\Lambda_2$, and we write $[\Lambda]$ for the equivalence class of the lattice Λ. We say that two equivalence classes of lattices are* adjacent *if there exist representative lattices Λ_1 and Λ_2 respectively such that $\pi\Lambda_1 \subseteq \Lambda_2 \subseteq \Lambda_1$.*

Definition 4.2. *Let $\mathcal{B}_d$ be the flag simplicial complex whose 0-simplices are equivalence classes of lattices in K^d and whose 1-simplices correspond to adjacent equivalence classes. We call $\mathcal{B}_d$ the* affine building *of $SL_d(K)$.*

Example 4.3. *Consider the building $\mathcal{B}_2$ for $K = \mathbb{Q}_3$ the 3-adic numbers. In this case our valuation ring $R = \mathbb{Z}_3$ is the 3-adic integers and our uniformizer π is 3. The affine building $\mathcal{B}_2$ is the infinite tree with every vertex having degree 4 in Figure 4.1.*

Definition 4.4. *Let $M = (v_1, \ldots, v_n)$ be a $d \times n$ matrix over K with columns $v_1, \ldots, v_n$ spanning K^d as a K-vector space, where $n > d$. The* membrane $[M]$ *of M is the collection of all lattice classes of the form $R\{\pi^{u_1}v_1, \ldots, \pi^{u_n}v_n\}$ for $u_i \in \mathbb{Z}$. If M is an invertible square $d \times d$ matrix, we call the corresponding membrane $[M]$ an* apartment.

Lemma 4.5 ([56], Lemma 4.13)**.** *Let M be a rank d matrix over K of size $d \times n$ with $n > d$. Then the membrane $[M]$ is the union of all apartments spanned by $d \times d$ invertible submatrices of M.*

Example 4.6. *Let $K = \mathbb{Q}_2$, and consider the rank-2 building $\mathcal{B}_2$ over K. This is an infinite tree where every vertex has degree 3. Within $\mathcal{B}_2$, the matrix $\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \end{pmatrix}$ defines the membrane*

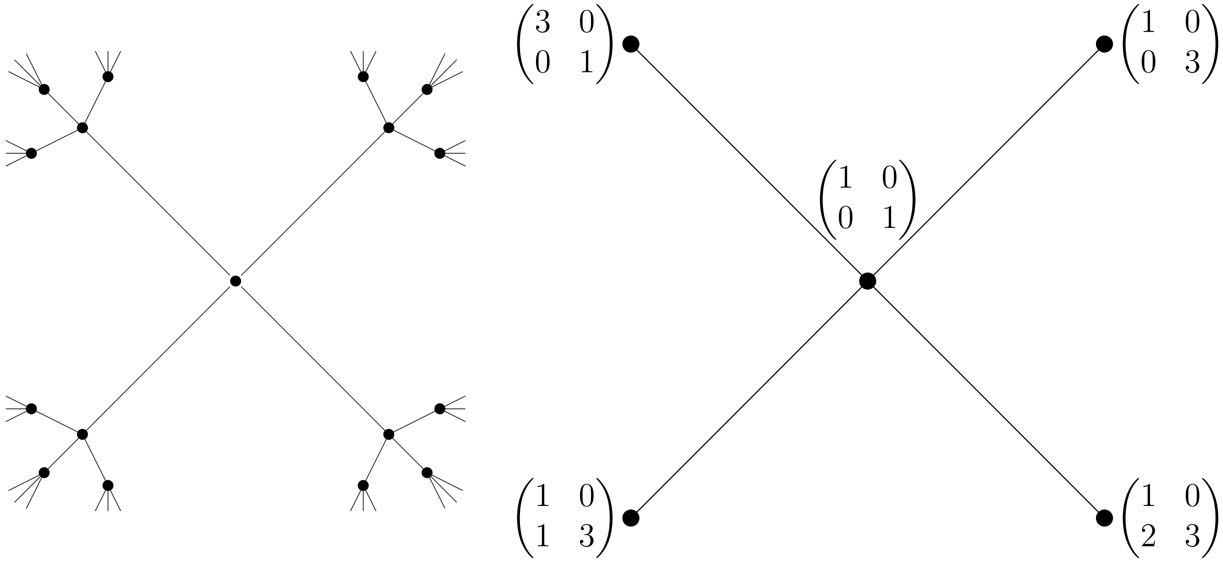

$\begin{pmatrix}3 & 0\\0 & 1\end{pmatrix}$
$\begin{pmatrix}1 & 0\\0 & 3\end{pmatrix}$
$\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}$
$\begin{pmatrix}1 & 0\\1 & 3\end{pmatrix}$
$\begin{pmatrix}1 & 0\\2 & 3\end{pmatrix}$

Figure 4.1: Left: the building $\mathcal{B}_2$ for $K = \mathbb{Q}_3$. Right: the star of the identity in this building.

in Figure 4.2, a simplicial subtree of the building consisting of three infinite paths emanating from a vertex of degree 3.

4.3 Min-convex hulls

Definition 4.7. *If Λ_1 and Λ_2 are lattices, then their intersection $\Lambda_1 \cap \Lambda_2$ is also a lattice. We say that a collection of lattice classes is* min-convex *or just* convex *if it is closed under taking equivalence classes of finite intersections of representative lattices.*

Given lattices $\Lambda_1, \ldots, \Lambda_s$, we call their min-convex hull *or* convex hull $\operatorname{conv}(\Lambda_1, \ldots, \Lambda_s)$ *the smallest convex set containing their lattice classes. We can similarly define the convex hull of an infinite collection of lattices. In addition, given invertible matrices $M_1, \ldots, M_s$, we write* $\operatorname{conv}(M_1, \ldots, M_s)$ *for the convex hull* $\operatorname{conv}(\Lambda_1, \ldots, \Lambda_s)$ *where each Λ_i is the lattice spanned by the columns of M_i.*

Remark 4.8. *There is another notion of convexity in [51] called* max-convexity*, which arises by considering sums of lattices instead of intersections. The duality functor $\Lambda \mapsto \Lambda^* = Hom_R(\Lambda, R)$ switches sums and intersections, so via this map the max-convex hull $maxconv(\Lambda_1, \ldots, \Lambda_s)$ is isomorphic to* $\operatorname{conv}(\Lambda_1^*, \ldots, \Lambda_s^*)$*. In particular, we may restrict our attention to convex hulls, and everything that follows can easily be translated to the language of max-convexity.*

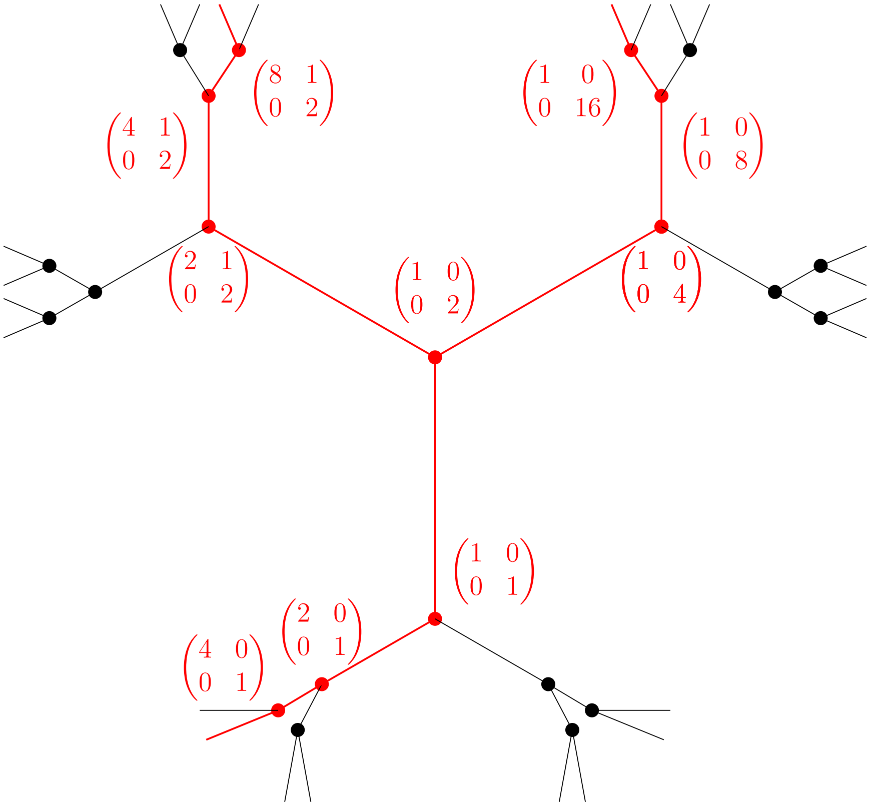

$\begin{pmatrix}8&1\\0&2\end{pmatrix}$
$\begin{pmatrix}1&0\\0&16\end{pmatrix}$
$\begin{pmatrix}4&1\\0&2\end{pmatrix}$
$\begin{pmatrix}1&0\\0&8\end{pmatrix}$
$\begin{pmatrix}2&1\\0&2\end{pmatrix}$
$\begin{pmatrix}1&0\\0&2\end{pmatrix}$
$\begin{pmatrix}1&0\\0&4\end{pmatrix}$
$\begin{pmatrix}1&0\\0&1\end{pmatrix}$
$\begin{pmatrix}2&0\\0&1\end{pmatrix}$
$\begin{pmatrix}4&0\\0&1\end{pmatrix}$

Figure 4.2: A membrane, in red, contained in the building $\mathcal{B}_2$ over $K = \mathbb{Q}_2$.

Lemma 4.9. *Let* $\Lambda_1, \ldots, \Lambda_s$ *be lattices. Then*

$$\operatorname{conv}(\Lambda_1, \ldots, \Lambda_s) = \bigcup_{\Lambda' \in \operatorname{conv}(\Lambda_2, \ldots, \Lambda_s)} \operatorname{conv}(\Lambda_1, \Lambda').$$

Proof. Pick any class V in $\operatorname{conv}(\Lambda_1, \ldots, \Lambda_s)$ with representative $\pi^{a_1}\Lambda_1 \cap \pi^{a_2}\Lambda_2 \cap \cdots \cap \pi^{a_s}\Lambda_s$. Clearly $\Lambda' = \pi^{a_2}\Lambda_2 \cap \cdots \cap \pi^{a_s}\Lambda_s$ satisfies $[\Lambda'] \in \operatorname{conv}(\Lambda_2, \ldots, \Lambda_s)$, so $V \in \operatorname{conv}(\Lambda_1, \Lambda')$. Conversely, fix a lattice $\Lambda' = \pi^{a'_2}\Lambda_2 \cap \cdots \cap \pi^{a'_s}\Lambda_s$ representing a class in $\operatorname{conv}(\Lambda_2, \ldots, \Lambda_s)$. Any class V in $\operatorname{conv}(\Lambda_1, \Lambda')$ has a representative of the form $\pi^{a_1}\Lambda_1 \cap \pi^{b}\Lambda' = \pi^{a_1}\Lambda_1 \cap \pi^{a_2}\Lambda_2 \cap \cdots \cap \pi^{a_s}\Lambda_s$, where $a_i = b + a'_i$ for $i = 2, \ldots, s$. In particular, V is certainly in $\operatorname{conv}(\Lambda_1, \ldots, \Lambda_s)$. □

The following result was originally stated in Faltings's paper on matrix singularities [27]. For completeness we provide an easy proof.

Lemma 4.10. *Let* $\Lambda_1, \ldots, \Lambda_s$ *be lattices representing equivalence classes in* $\mathcal{B}_d$. *Then the convex hull* $\operatorname{conv}(\Lambda_1, \ldots, \Lambda_s)$ *is finite.*

Proof. Any class in $\operatorname{conv}(\Lambda_1, \Lambda_2)$ has a representative of the form $\Lambda_1 \cap \pi^a\Lambda_2$ for some $a \in \mathbb{Z}$, so it suffices to consider such intersections for all possible choices of a. When $a \gg$

0 we know that $\Lambda_1 \supseteq \pi^a\Lambda_2$, so that $\Lambda_1 \cap \pi^a\Lambda_2 = \pi^a\Lambda_2$. When $a \ll 0$ we know $\Lambda_1 \subseteq \pi^a\Lambda_2$, so that $\Lambda_1 \cap \pi^a\Lambda_2 = \Lambda_1$. Hence the convex hull of two lattices is finite. Lemma 4.9 describes $\operatorname{conv}(\Lambda_1, \ldots, \Lambda_s)$ as the union of convex hulls of the form $\operatorname{conv}(\Lambda_1, \Lambda')$, where $\Lambda' \in \operatorname{conv}(\Lambda_2, \ldots, \Lambda_s)$. In particular, $\operatorname{conv}(\Lambda_1, \ldots, \Lambda_s)$ is a finite union of finite sets by induction and hence is itself finite. □

Example 4.11. *The proof of Lemma 4.10 shows that the convex hull of two lattice classes is a path between the two points in the building. We illustrate this fact in the following example. Let $K = \mathbb{Q}_2$, and consider the rank-3 building $\mathcal{B}_3$ over K. Take the two matrices*

$$M_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ and } M_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{4} & 0 \\ 0 & 0 & 8 \end{pmatrix},$$

which correspond to lattices Λ_1 and Λ_2 respectively. We see that $\Lambda_1 \cap 2^{-3}\Lambda_2 = \Lambda_1$ and $\Lambda_1 \cap 2^2\Lambda_2 = 2^2\Lambda_2$. The convex hull $\operatorname{conv}(\Lambda_1, \Lambda_2)$ is a path between $[\Lambda_1]$ and $[\Lambda_2]$ as depicted in Figure 4.3, with a representative lattice labeled above each vertex.

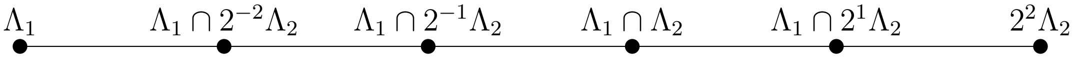

Λ_1
$\Lambda_1 \cap 2^{-2}\Lambda_2$
$\Lambda_1 \cap 2^{-1}\Lambda_2$
$\Lambda_1 \cap \Lambda_2$
$\Lambda_1 \cap 2^{1}\Lambda_2$
$2^2\Lambda_2$

Figure 4.3: The convex hull of Λ_1 and Λ_2 in $\mathcal{B}_3$, labeled by representative lattices.

Remark 4.12. *As Lemma 4.10 and Example 4.11 show, the convex hull $\operatorname{conv}(\Lambda_1, \Lambda_2)$ forms a path in the building between $[\Lambda_1]$ and $[\Lambda_2]$. In fact, this path can be easily checked to be a geodesic under the coweight metric of [55], where distances between points of the building are vectors in the dominant Weyl chamber of $SL_d(K)$. It is well-known that affine buildings such as $\mathcal{B}_d$ have a metric property called being* CAT(0)*, which implies that unique geodesics exist under a different metric that is locally Euclidean in each apartment. The length of such a CAT(0)-geodesic is the 2-norm of the coweight distance between $[\Lambda_1]$ and $[\Lambda_2]$ [32, p. 1884], so it can easily be computed from $\operatorname{conv}(\Lambda_1, \Lambda_2)$.*

Because convex hulls are finite, it is natural to ask how to compute them. In fact, the building $\mathcal{B}_d$ and membranes have an innate tropical structure which can be exploited for this purpose.

In effect, membranes are just standard triangulations of tropicalized linear spaces.

Theorem 4.13 ([56], Theorem 4.15)**.** *Let $M = (v_1, \ldots, v_n)$ be a $d \times n$ matrix of rank d over K and let L be its associated tropical linear space. Then there is a simplicial complex isomorphism Ψ_M between the membrane $[M]$ and the standard triangulation of L,*

$$\Psi_M(R\{\pi^{-u_1}v_1, \ldots, \pi^{-u_n}v_n\}) := pr_L((u_1, \ldots, u_n)),$$

sending a lattice $R\{\pi^{-u_1}v_1, \ldots, \pi^{-u_n}v_n\}$ to the projection onto L of the point $(u_1, \ldots, u_n) \in \mathbb{TP}^{n-1}$.

As a first illustration of this theorem, note that Figures 4.2 and 1.4 are isomorphic as simplicial complexes, where the tropical linear space in Figure 1.4 inherits the standard triangulation of $\mathbb{TP}^2$ as a tropical lattice polytope. Both are trees comprising three infinite branches stemming from a single node.

Example 4.14. *If our matrix M is square, so that its membrane $[M]$ is actually an apartment in the building, then Ψ_M describes a simplicial complex isomorphism between the apartment and the* tropical projective torus $\mathbb{R}^n/\mathbb{R}\mathbf{1}$.

Example 4.15. *Keep the notation of Theorem 4.13. Rincón [75] describes a local structure of any tropical linear space L with Plücker vector p, in which a basis σ of the underlying matroid yields a* local tropical linear space *defined by*

$$L_\sigma = \{u \in L : p(\sigma) - \sum_{i\in\sigma} u_i \leq p(\tau) - \sum_{j\in\tau} u_j \,\forall \text{ bases } \tau\}.$$

These local tropical linear spaces L_σ are isomorphic to Euclidean space $\mathbb{R}^{n-1}$, are contained in L, and together form a non-disjoint cover of L.

The covering of a membrane by its apartments derives from this local structure of tropical linear spaces. In particular, let σ describe a basis of the matroid of M, so that the $d \times d$ matrix M_σ with columns indexed by σ is invertible. Then the linearity of the determinant over column sums implies that the apartment $[M_\sigma]$ is mapped by Ψ_M to the local tropical linear space L_σ.

Given any membrane $[M]$ represented by a $d \times n$ matrix $M = (f_1, \dots, f_n)$, there is a *retraction* r_M of the entire building $\mathcal{B}_d$ onto $[M]$, which restricts to the identity on $[M]$ itself:

$$r_M : \Lambda \mapsto (\Lambda \cap K\{f_1\}) + \cdots + (\Lambda \cap K\{f_n\}).$$

We may use this map to describe a tropical structure for convex hulls.

Theorem 4.16 ([51], Proposition 22)**.** *Let M be a $d \times n$ matrix of rank d over K, $[M]$ its corresponding membrane, and L its corresponding tropical linear space. Also let $\Lambda_1, \dots, \Lambda_s$ be lattices corresponding to points in $\mathcal{B}_d$. The following two simplicial complexes coincide:*

$$r_M(\mathrm{conv}(\Lambda_1, \dots, \Lambda_s)) \subseteq [M],$$

$$\mathrm{tconv}(\Psi_M(r_M(\Lambda_1)), \dots, \Psi_M(r_M(\Lambda_s))) \subseteq L.$$

In particular, if $[M]$ contains the convex hull of $\Lambda_1, \dots, \Lambda_s$, then the retraction map acts as the identity, and the convex hull $\mathrm{conv}(\Lambda_1, \dots, \Lambda_s)$ is isomorphic to the standard triangulation of a tropical polytope. This suggests an approach for computing convex hulls in $\mathcal{B}_d$ as follows:

Algorithm 4.17 (Convex hull computation)**.**

```
Require: $d \times d$ invertible matrices $M_1, \dots, M_s$ over $K$ whose columns are bases for lattices $\Lambda_1, \dots, \Lambda_s$
Ensure: $\mathrm{conv}(\Lambda_1, \dots, \Lambda_s) \subseteq \mathcal{B}_d$
1: $[M] \leftarrow$ a membrane containing $\mathrm{conv}(\Lambda_1, \dots, \Lambda_s)$
2: for all $i \in \{1, \dots, s\}$ do
3:     $P_i \leftarrow \Psi_M(r_M(\Lambda_i))$
4: end for
5: $X \leftarrow \mathrm{tconv}(P_1, \dots, P_s)$
6: return $X$
```

Note that given a lattice Λ_i represented by a matrix M_i, we can compute the image $\Psi_M(r_M(\Lambda_i))$ in $\mathbb{TP}^{n-1}$ simply by taking the tropical row sum of the matrix $\mathrm{val}(M_i^{-1}M)$, by [51, Lemma 21].

Example 4.18. *Let $K = \mathbb{Q}_5$ and consider the matrices*

$$M_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, M_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{5} & 0 \\ 0 & 0 & \frac{1}{125} \end{pmatrix}, M_3 = \begin{pmatrix} 5 & 625 & 150 \\ 0 & 25 & 1 \\ 0 & 0 & \frac{1}{5} \end{pmatrix},$$

whose convex hull is shown in Figure 4.4.

Let M be the matrix

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

The membrane $[M]$ turns out to contain $\mathrm{conv}(M_1, M_2, M_3)$. Running through Algorithm 4.17 with this membrane yields the following tropical matrix,

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 \\ -1 & -2 & 1 & -1 \end{pmatrix},$$

whose rows or columns span the tropical convex hull in Figure 4.5. Note that tropical polytopes are self-dual, i.e. the columns and rows of any tropical matrix span isomorphic tropical polytopes [22, Theorem 1].

From this tropical polytope we can construct representatives for any of the lattice classes in Figure 4.4. For example, consider the central lattice point $(0, 1, -1)$ with six neighbors. By Equation (14) in [22], this lattice point in the column-span of our tropical matrix corresponds to $(-1, -1, 0, 0)$ in the row-span. In turn, this point corresponds to the class of the lattice

$$\mathbb{Z}_5 \left\{ 5^1 \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, 5^1 \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, 5^0 \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, 5^0 \cdot \begin{pmatrix} 0 \\ 5 \\ 1 \end{pmatrix} \right\}.$$

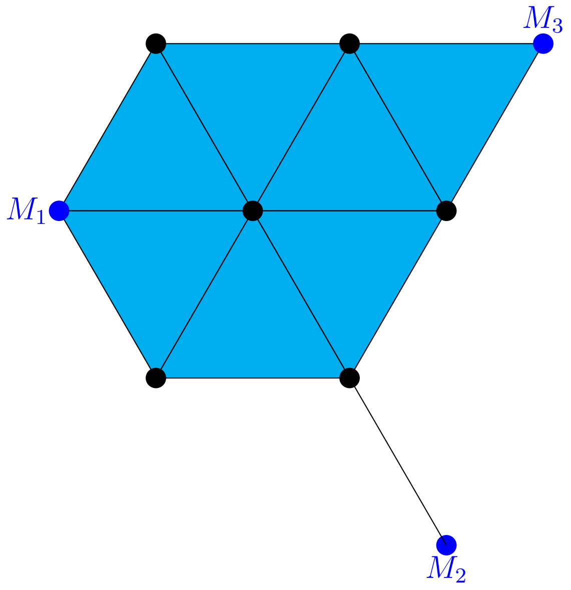


Figure 4.4: The convex hull of the matrices M_1, M_2, and M_3 in $\mathcal{B}_3$.

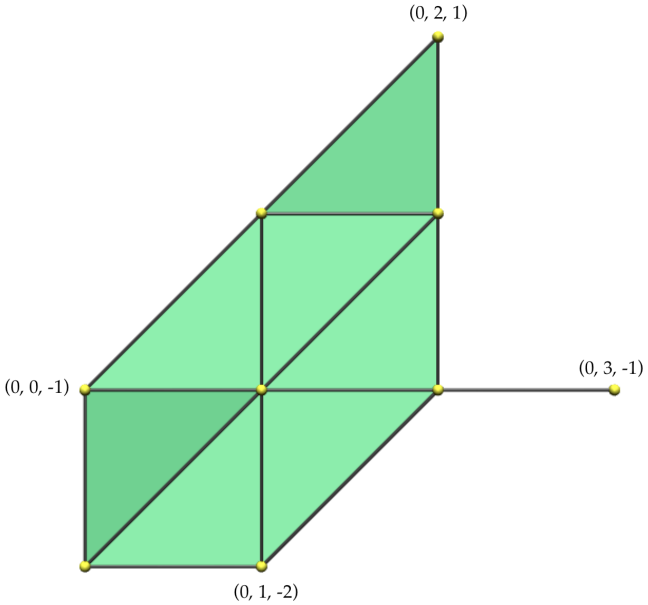


Figure 4.5: A tropical polytope isomorphic as a simplicial complex to $\operatorname{conv}(M_1, M_2, M_3)$, with coordinates of spanning vertices labeled.

Of course, Algorithm 4.17 requires an *enveloping membrane* of $\Lambda_1, \ldots, \Lambda_s$: a membrane containing the convex hull of $\Lambda_1, \ldots, \Lambda_s$. Without such a membrane the tropical polytope produced by Algorithm 4.17 need not be isomorphic to our original convex hull. Note that membranes need not be convex [51, Example 5], so it is not sufficient to simply pick a membrane containing the spanning lattice classes.

No bounded-time algorithm for computing an arbitrary enveloping membrane has thus far been described. A procedure for computing such an enveloping membrane is described in [51], but it is often impractical: it relies on the computation of each individual element of the convex hull, while expanding a starting membrane to contain each element whenever necessary. In Section 4.5 we will describe an improved algorithm with bounded complexity in d and s, allowing us to algorithmically realize any convex hull as a tropical polytope.

Remark 4.19. *Our notion of convexity was originally introduced by Faltings [27], who noted that configurations Γ of vertices in $\mathcal{B}_d$ correspond to certain schemes $M(\Gamma)$ called* Mustafin varieties *or* Deligne schemes *over the spectrum of a DVR. In the arithmetic setting, these Mustafin varieties function as local models of Shimura varieties [73]. The special fiber of $M(\Gamma)$ generally has many singularities, but replacing Γ with its convex hull Γ' yields a regular Mustafin variety $M(\Gamma')$ with a dominant morphism $M(\Gamma') \to M(\Gamma)$, such that the irreducible* *components of the special fiber of $M(\Gamma')$ intersect transversally [12, Lemma 2.4 and Theorem 2.10]. In this way, our fully-specified version of Algorithm 4.17, derived in Section 4.5, allows for the explicit resolution of singularities of Mustafin varieties.*

4.4 Simultaneously-adaptable bases

In this section we review a classical result on lattices over valued fields, following [1, Section 6.9], and describe its relevance to our setting of convex hulls in affine buildings. Note that in what follows we say *monomial matrix* to refer to any matrix A supported on a permutation matrix: i.e., there exists a permutation σ such that $A_{ij} \neq 0$ if and only if $j = \sigma(i)$.

Algorithm 4.20 (Simultaneously adaptable basis for two lattices)**.**

```
Require: $M_1, M_2$ $d \times d$ invertible matrices over $K$ whose columns are bases for lattices $\Lambda_1$
    and $\Lambda_2$ in $\mathcal{B}_d$
Ensure: Invertible matrix $A$, monomial matrix $\Delta$ such that the columns of $A$ and $A\Delta$ are
    bases for $\Lambda_1$ and $\Lambda_2$, respectively
$B_1 \leftarrow M_1$
$C_1 \leftarrow M_2$
for all $i \in \{1, \dots, d-1\}$ do
    $N_i \leftarrow B_i^{-1} C_i$
    $n_i \leftarrow$ entry of minimal valuation in $N_i$ not equal to $n_1, \dots, n_{i-1}$
    $L_i \leftarrow$ the matrix such that $L_i N_i$ is obtained from $N_i$ by eliminating all other elements
        in the column of $n_i$
    $R_i \leftarrow$ the matrix such that $L_i N_i R_i$ is obtained from $L_i N_i$ by eliminating all other
        elements from the row of $n_i$
    $B_{i+1} \leftarrow B_i L_i^{-1}$
    $C_{i+1} \leftarrow C_i R_i$
end for
$A \leftarrow B_d$
$\Delta \leftarrow B_d^{-1} C_d$
return $A, \Delta$
```

Lemma 4.21. *Let M_1 and M_2 be $d \times d$ invertible matrices over K for lattices Λ_1 and Λ_2 in $\mathcal{B}_d$. Then Algorithm 4.20 correctly returns a basis A for Λ_1 and a monomial matrix Δ such that $A\Delta$ is a basis for Λ_2.*

Proof. Because n_i is chosen to be of minimal valuation in N_i, each L_i and R_i will be matrices in $SL_d(R)$. It follows that the new matrices B_{i+1} and C_{i+1} will be bases for Λ_1 and Λ_2 if B_i and C_i are, with base change matrix $L_i N_i R_i$. In particular, after $d-1$ steps of the for-loop, the matrix $\Delta = L_{d-1}N_{d-1}R_{d-1}$ will have $d-1$ distinct entries which are uniquely nonzero in their respective rows and columns. Hence Δ is a monomial matrix, as desired. □

Definition 4.22. *We call the output A in Algorithm 4.20 a* simultaneously adaptable basis *(*SA-basis*) for Λ_1 and Λ_2.*

Algorithm 4.20 is a demonstration of the building-theoretic fact that any two lattices lie in a common apartment. In general, there should be many distinct apartments containing any two given points. Indeed, the SA-basis A obtained above depends not only on the lattices Λ_1 and Λ_2 but on our original choice of bases, which lattice we designate as Λ_1, and how we break ”ties” between elements of minimal valuation when choosing pivots. In what follows, we break ties between potential pivots by picking the option in the leftmost column, then topmost row.

Because apartments are convex, we have the following fact:

Corollary 4.23. *Pick two lattice classes represented by Λ_1 and Λ_2 in $\mathcal{B}_d$, and let A be a SA-basis for the two lattices. Then the apartment $[A]$ contains the convex hull* $\mathrm{conv}(\Lambda_1, \Lambda_2)$.

Lemma 4.24. *Let M_1 and M_2 be two invertible $d \times d$ matrices representing lattices Λ_1 and Λ_2, and let Γ be any diagonal matrix with $M_2' = M_2\Gamma$. Let N_i and N_i' be the base-change matrices at the ith step of Algorithm 4.20 executed with the pairs (M_1, M_2) and (M_1, M_2') as input respectively, n_i and n_i' the chosen pivots of least valuation at step i, and so on. If k is a positive integer such that the positions of the pivots n_i and n_i' agree for all i up to $k-1$, then $L_{k-1} = L_{k-1}'$ and $N_k' = N_k\Gamma$.*

Proof. We prove the result by induction, noting that the base case $k = 1$ follows trivially. Suppose that the first $k-1$ pivots are the same for the two algorithm executions. Because the first $k-2$ pivots are the same, by the inductive hypothesis we have that $N_{k-1}' = N_{k-1}\Gamma$. In particular, the ratio of any two entries in the same column is the same for N_{k-1} and N_{k-1}'. Now since the $k-1$st pivot position is also the same, the row operations to obtain N_k and N_k' from N_{k-1} and $N_{k-1}' = N_{k-1}\Gamma$ agree as well, so that $L_{k-1} = L_{k-1}'$. Next the column operations necessary to clear the rows of two pivots may differ, but in both executions we eliminate using a column which has no other nonzero entries. It follows that $N_k' = N_k\Gamma$, as desired. □

Corollary 4.25. *Keep the setup of Lemma 4.24 above, and let A be the basis of Λ_1 produced by Algorithm 4.20. If all pivot positions of Algorithm 4.20 are the same for the two inputs (M_1, M_2) and (M_1, M_2'), then the lattice class $[\Lambda_2']$ for M_2' is contained in the apartment $[A]$.*

Proof. Because all pivots are the same, Lemma 4.24 implies that $L_i = L_i'$ for all i. This means the final basis for Λ_1 produced by both executions of the algorithm is $A = L_dL_{d-1}\dots L_1M_1$. In particular, we have that $[A]$ contains both $[\Lambda_2]$ and $[\Lambda_2']$. □

In fact, a stronger statement holds: it suffices to assume that corresponding pairs of pivots share the same pivot columns.

Corollary 4.26. *Retain the setup of Lemma 4.24 above, and let A be the basis of Λ_1 produced by Algorithm 4.20. Suppose that for all i, the ith pivots of Algorithm 4.20 for the two inputs (M_1, M_2) and (M_1, M_2') appear in the same pivot column. Then the lattice class $[\Lambda_2']$ for M_2' is contained in the apartment $[A]$.*

Proof. We prove by induction on the ith pivot that all pivots are actually in the same positions; the result then follows by Corollary 4.25. When $i = 1$, the two base-change matrices are $N_1 = M_1^{-1}M_2$ and $N_1' = M_1^{-1}M_2\Gamma$. By assumption, the first pivots of the two algorithm executions share the same pivot column. We note that scaling a column does not change the position of the minimal-valuation column element. It follows that the first pivots are in the same position.

Now suppose that the first $i-1$ pivots share the same positions, so that by Lemma 4.24 we have $N_i' = N_i\Gamma$. Again, by assumption the ith pivots appear in the same column, and the same argument as the base case implies the ith pivots will be in the same position as well. □

We can apply Corollary 4.26 to cover the convex hull of an apartment and a lattice class.

Lemma 4.27. *Let A be an invertible matrix defining an apartment $[A]$ and M a basis for a lattice Λ whose class is not in $[A]$. Then $\operatorname{conv}([\Lambda], [A])$ can be covered with $d!$ different apartments.*

Proof. It suffices to find a collection of apartments covering $\operatorname{conv}(M, A\Gamma)$ for all diagonal matrices Γ. Fix such a Γ, and let B be the basis for M produced by Algorithm 4.20 applied to the pair $(M, A\Gamma)$. Let Γ' be any other diagonal matrix such that Algorithm 4.20 applied to $(M, A\Gamma')$ uses the same sequence of pivot columns. Then Corollary 4.26 implies that $[B]$ contains $\operatorname{conv}(M, A\Gamma')$ as well. In other words, we need only one apartment for each sequence of pivot columns. Since each pivot must appear in a different column, there are $d!$ different sequences of pivot columns, and so we need at most $d!$ such apartments to cover the entirety of $\operatorname{conv}([\Lambda], [A])$. □

Remark 4.28. *We note the similarity of Lemma 4.27 with [45, Lemma 6.3], which states that any apartment A in any building can be covered by the union of Weyl chambers based at some other fixed point z with equivalence class in ∂A, the spherical apartment at infinity corresponding to A. We expect that Lemma 4.27 is an explicit analogue of this result in our specialized setting, where ∂A is isomorphic to the symmetric group S_d on d elements, in which each Weyl chamber is replaced by a suitable apartment containing it to ensure the convex hull of z and A is also covered.*

4.5 Constructing enveloping membranes

In this section we combine the results of the previous section to solve the problem left open in Algorithm 4.17. Namely, we present an algorithm to compute an enveloping membrane of a finite set of lattices. This allows us to realize convex hulls in the building as tropical polytopes.

Algorithm 4.29 (List of apartments covering a convex hull).

```
Require: base matrices $B_1, \ldots, B_s$ for lattices $\Lambda_1, \ldots, \Lambda_s$
Ensure: A set of apartments covering $\operatorname{conv}(\Lambda_1, \ldots, \Lambda_s)$
if $s = 2$ then
    return SA-basis of $\Lambda_1$ and $\Lambda_2$ via Algorithm 4.20
end if
$L_{s-1} \leftarrow$ set of apartments covering $\operatorname{conv}(\Lambda_2, \ldots, \Lambda_s)$ via Algorithm 4.29
$L_s \leftarrow \emptyset$
for all $[A] \in L_{s-1}$ do
    $L_A \leftarrow$ set of apartments covering $\operatorname{conv}(\Lambda_1, [A])$ as in Remark 4.31
    $L_s \leftarrow L_s \cup L_A$
end for
return $L_s$
```

Theorem 4.30. *Let $M_1, \ldots, M_s$ represent s lattices $\Lambda_1, \ldots, \Lambda_s$ in $\mathcal{B}_d$. Then Algorithm 4.29 correctly computes a list of apartments L_s such that each lattice class $[\Lambda] \in \operatorname{conv}(\Lambda_1, \ldots, \Lambda_s)$ is contained in $[A]$ for some $[A] \in L_s$. Furthermore, L_s has size at most $(d!)^{s-2}$.*

Of course this theorem and Lemma 4.5 together imply that Algorithm 4.29 can be used to compute an enveloping membrane for $\Lambda_1, \ldots, \Lambda_s$. We simply concatenate all the matrices in the output L_s.

Proof. If the algorithm is correct, then L_{s-1} contains at most $(d!)^{s-3}$ apartments by induction. Since L_A has size at most $d!$ by Lemma 4.27, L_s has size at most $(d!)^{s-2}$.

It remains to prove correctness. By Lemma 4.9, any lattice class $[\Lambda]$ in $\operatorname{conv}(\Lambda_1, \ldots, \Lambda_s)$ is contained in $\operatorname{conv}(\Lambda_1, \Lambda')$ for some $[\Lambda'] \in \operatorname{conv}(\Lambda_2, \ldots, \Lambda_s)$. There exists some $[A] \in L_{s-1}$ such that $[\Lambda'] \in [A]$, and so $[\Lambda] \in \operatorname{conv}(\Lambda_1, [A])$. In particular, there is some $[B] \in L_A$ such that $[\Lambda] \in [B]$. □

Remark 4.31. *The crucial part of Algorithm 4.29 is computing the set L_A of apartments covering $\operatorname{conv}(\Lambda_1, [A])$. Recall from Lemma 4.27 that this set is indexed by permutations in S_d. We sketch here how to compute the apartment corresponding to the identity permutation; all other apartments can be computed very similarly.*

Let $\Gamma = \operatorname{diag}(\pi^{a_1}, \ldots, \pi^{a_d})$ be a diagonal matrix with integers $a_1, \ldots,$ $a_d \in \mathbb{Z}$. First choose Γ to be any diagonal matrix such that the first pivot for the first base change matrix $M_1^{-1}A\Gamma$ is in the first column. Next we compute the second base-change matrix; by decreasing both a_1 and a_2 by a large enough common value, Lemma 4.24 guarantees that the first pivot will still be in the first column, and that the second pivot will appear in the second column. We next compute the third base-change matrix by reducing a_1, a_2, and a_3 all by some large enough value, and so on.

Remark 4.32. *We present in the next section a more efficient algorithm for the $s = 3$ case, Algorithm 4.38, needing only 2^d apartments to cover the convex hull instead of $d!$. Because Algorithm 4.29 is inductive on s, Algorithm 4.38 can be used for the $s = 3$ case, providing a slightly better overall bound of $2^d \cdot (d!)^{s-3}$ apartments needed to cover the convex hull of s lattices.*

Corollary 4.33. *Let $\Lambda_1, \ldots, \Lambda_s$ be lattices in K^d. Then their convex hull* $\operatorname{conv}(\Lambda_1, \ldots, \Lambda_s)$ *is isomorphic to a tropical polytope in $\mathbb{TP}^N$ where $N \leq d \cdot 2^d \cdot (d!)^{s-3} - 1$.*

Proof. Remark 4.32 implies that a matrix M with at most $d \cdot 2^d \cdot (d!)^{s-3}$ columns generates a membrane $[M]$ containing $\operatorname{conv}(\Lambda_1, \ldots, \Lambda_s)$. Algorithm 4.17 then realizes the convex hull as a tropical polytope in a tropical projective space of dimension at most $d \cdot 2^d \cdot (d!)^{s-3} - 1$. □

Corollary 4.34. *Let $\Lambda_1, \ldots, \Lambda_s$ be lattices in K^d. Then their convex hull* $\operatorname{conv}(\Lambda_1, \ldots, \Lambda_s)$ *is isomorphic to a tropical polytope spanned by at most $d \cdot 2^d \cdot (d!)^{s-3}$ points in $\mathbb{TP}^{s-1}$.*

Proof. This follows directly from Corollary 4.33 and the self-duality of tropical polytopes. □

Corollary 4.35. *Let $\Lambda_1, \ldots, \Lambda_s$ be lattices in K^d. Let $[M]$ be the enveloping membrane for* $\operatorname{conv}(\Lambda_1, \ldots, \Lambda_s)$ *computed by concatenating the apartments from Algorithm 4.29. Then Λ_1 is mapped to the origin by Ψ_M in Algorithm 4.17.*

Proof. There is another representation of the building $\mathcal{B}_d$, which describes the vertices as certain maps $N : K^d \to \mathbb{R} \cup \{\infty\}$ called *integral additive norms* [51, pp. 189-190]. We can easily pass between these two descriptions of the building in terms of lattice classes and integral additive norms. If Λ is a lattice represented by a matrix M, then the corresponding integral additive norm is defined by

$$N_\Lambda(v) = \max\{u \in \mathbb{Z} : \pi^{-u} v \in \Lambda\}.$$

Write $M = (v_1, \ldots, v_n)$. By [51, Lemma 21], the image of Λ_1 under the map of Theorem 4.16 is $(N_{\Lambda_1}(v_1), \ldots, N_{\Lambda_1}(v_n))$, where N_{Λ_1} is the integral additive norm corresponding to Λ_1. But clearly $N_{\Lambda_1}(v_i) = 0$ for each i, since each v_i is an element for a basis for Λ_1. □

Viewed in the dual setting of Corollary 4.34, Corollary 4.35 implies that our algorithm places us in the affine chart of $\mathbb{TP}^{s-1}$ where the first coordinate is zero.

Example 4.36. *Another common discrete valued field is $K = \mathbb{C}((t))$, the field of complex formal Laurent series. In this case, we have $R = \mathbb{C}[\![t]\!]$ the ring of complex formal power series, $\pi = t$ a uniformizer, and $k = \mathbb{C}$ the residue field. Consider the following four* 3×3 *matrices over K:*

$$M_1 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & t & t^2 \\ 1 & t^{-2} & t \end{pmatrix}, M_2 = \begin{pmatrix} 1 & 1 & 1 \\ t & t^2 & t^3 \\ t^{-2} & t & t^5 \end{pmatrix}, M_3 = \begin{pmatrix} 1 & 1 & 1 \\ t^2 & t^3 & t^4 \\ t & t^5 & t^8 \end{pmatrix}, M_4 = \begin{pmatrix} 1 & 1 & 1 \\ t^3 & t^4 & t^5 \\ t^5 & t^8 & t^{12} \end{pmatrix}.$$

These are the contiguous maximal submatrices of

$$M = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & t & t^2 & t^3 & t^4 & t^5 \\ 1 & t^{-2} & t & t^5 & t^8 & t^{12} \end{pmatrix},$$

so the corresponding lattice classes certainly all lie in the membrane $[M]$*. An optimist could suppose that* $[M]$ *were in fact an enveloping membrane for the convex hull of our four matrices. Running through Algorithm 4.17 with the membrane* $[M]$ *yields the following tropical matrix:*

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 & 0 & 0 \\ -3 & -4 & 0 & 0 & 0 & 0 \\ -6 & -8 & -5 & 0 & 0 & 0 \end{pmatrix}.$$

The columns of this matrix span the tropical polytope P*, visualized using Polymake in Figure 4.6. Its standard triangulation contains 18 vertices, 32 edges, and 15 triangles.*

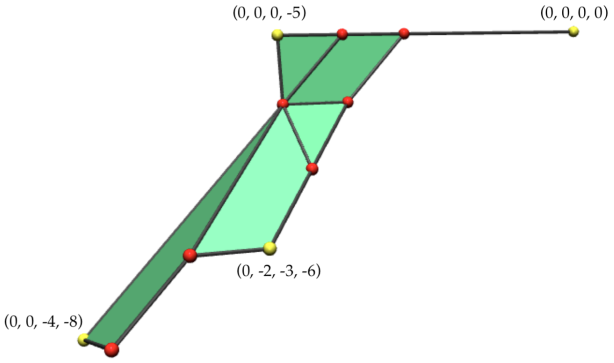


Figure 4.6: The tropical polytope P obtained by using the membrane $[M]$ for the lattices M_1, M_2, M_3, and M_4 with Algorithm 4.17. Points spanning the tropical convex hull are marked in yellow.

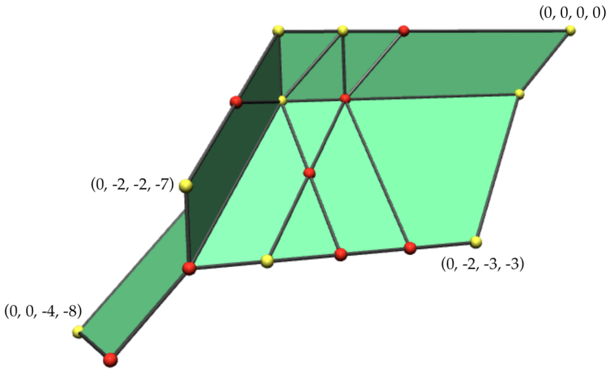


Figure 4.7: The tropical polytope P' whose standard triangulation is isomorphic to the convex hull of M_1, M_2, M_3, and M_4, with spanning vertices marked in yellow.

However, when we run Algorithm 4.29 in Polymake to compute an enveloping membrane for the convex hull $\operatorname{conv}(M_1, M_2, M_3, M_4)$*, we obtain a different matrix* M' *with 12 distinct columns. Executing Algorithm 4.17 using the membrane* $[M']$ *yields that* $\operatorname{conv}(M_1, M_2, M_3, M_4)$ *is isomorphic as a simplicial complex to the tropical polytope* P' *in Figure 4.7 spanned by*

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 & -2 & 0 & -2 & 0 & 0 & -2 & 0 & 0 \\ -3 & -4 & 0 & -1 & -2 & 0 & -2 & 0 & -1 & -3 & -1 & 0 \\ -6 & -8 & -5 & -5 & -7 & 0 & -7 & -4 & -1 & -3 & -5 & 0 \end{pmatrix}.$$

The standard triangulation of this polytope contains 29 lattice points, 67 edges, and 41 triangles. In particular, the convex hull of M_1, M_2, M_3, and M_4 is larger than the polytope P obtained via the membrane $[M]$, even though each lattice spanning the convex hull is trivially contained in $[M]$. In turn, this means that $[M]$ does not contain the convex hull $\operatorname{conv}(M_1, M_2, M_3, M_4)$*, demonstrating the fact that membranes are not convex.*

Example 4.37. *Again take $K = \mathbb{C}((t))$ the field of complex formal Laurent series. Let M_1 be the 4×4 identity matrix, M_2 be diagonal with entries $1, t^3, t^{-2}$, and t^{-2} respectively, and*

$$M_3 = \begin{pmatrix} t^{-3} - t^2 & 1 - t^2 & -t^{-2} + 1 & t^{-2} - t \\ t^2 - t^3 & -t^{-3} + t & 1 - t & 0 \\ 0 & -1 + t & t^{-3} - t^3 & t^{-3} - 1 \\ -t + t^2 & -t^{-1} + 1 & 0 & -t^{-1} + t^2 \end{pmatrix},$$

$$M_4 = \begin{pmatrix} 1 - t^3 & t^{-1} - 1 & 1 - t^2 & 1 - t^3 \\ t^{-3} - 1 & 1 - t & 1 - t^2 & 1 - t^3 \\ -t^{-3} + t & -t^{-2} + 1 & -t^{-3} + t^{-1} & -1 + t \\ t^{-3} - t^{-2} & -t^{-1} + 1 & -t^{-1} + 1 & t^{-1} - 1 \end{pmatrix}.$$

Concatenating the matrices produced by Algorithm 4.29 applied to M_1, M_2, M_3, and M_4 in Polymake gives a matrix M with 84 distinct columns. Using the corresponding membrane $[M]$ with Algorithm 4.17, we get a 4×84 matrix over the tropical numbers. After pruning duplicate columns, we obtain the following matrix whose tropical row or column span gives the polytope displayed in Figure 4.8. The triangulation of that polytope has 30 vertices, 95 edges, 102 triangles, and 36 tetrahedra.

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3 & -2 & -1 & -3 & -3 & 0 & 2 & 2 & 0 & -1 & -1 & -3 \\ 1 & 2 & 3 & 1 & 1 & 3 & 3 & 1 & 1 & 1 & 3 & 3 \\ 3 & 3 & 3 & 2 & 1 & 1 & 3 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

4.6 Convex triangles

Suppose that $s = 3$, so that we wish to compute a *convex triangle*: the convex hull of three lattice classes. This is relevant e.g. to [12, Section 4.6], which focuses on Mustafin varieties arising from convex triangles. In this case there exists a more efficient algorithm, taking advantage of the fact that $\operatorname{conv}(\Lambda_2, \Lambda_3)$ is just a path in the building. We now describe this improvement.

With some extra book-keeping, note that Algorithm 4.20 can output all of the following:

- an SA-basis A which is a basis for Λ_1,

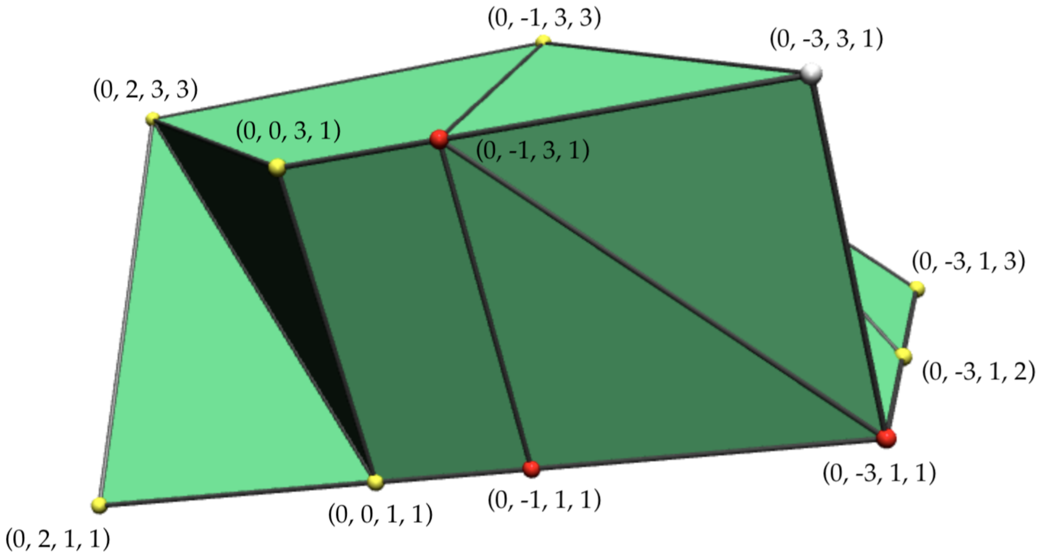


Figure 4.8: The 3-dimensional tropical polytope isomorphic to the convex hull of our matrices M_1, M_2, M_3, M_4, whose standard triangulation has f-vector (30, 95, 102, 36).

- a diagonal matrix $\Delta = \operatorname{diag}(\pi^{c_1}, \dots, \pi^{c_d})$ such that $A\Delta$ is a basis for Λ_2, where $c_1 \leq \cdots \leq c_d$ are integers in increasing order,
- all of the base change matrices $N_1, \dots, N_d$,
- and the positions $p_1, \dots, p_d$ of the pivots $n_1, \dots, n_d$.

We justify the existence of such an (A, Δ) pair. First, note that if a matrix M is a basis for a lattice Λ and P is any permutation matrix, then MP is also a basis for Λ, since the matrix multiplication amounts to a reordering of the basis vectors. It follows that the base-change matrix Δ produced by Algorithm 4.20 can always be taken to be diagonal, as otherwise we may multiply it on the right by an appropriate permutation matrix to make it so. We can therefore write $\Delta = \operatorname{diag}(\pi^{c_1}, \dots, \pi^{c_d})$. If $c_1 \leq \cdots \leq c_d$ does not hold, there exist permutation matrices P_1 and P_2 such that $P_1 \Delta P_2$ is diagonal and in valuation-sorted order. Replacing A with AP_1 and Δ with ΔP_2 yields a pair with the desired properties.

Algorithm 4.38 (Enveloping membrane for a convex triangle)**.**

```
Require: three $d \times d$ invertible matrices $M_1, M_2, M_3$ over $K$ whose columns are bases for
    lattices $\Lambda_1, \Lambda_2, \Lambda_3$ in $\mathcal{B}_d$
Ensure: A list $L$ of apartments covering $\operatorname{conv}(\Lambda_1, \Lambda_2, \Lambda_3)$.
  $L \leftarrow \emptyset$
```

```
$(A, \Delta = \mathrm{diag}(\pi^{c_1}, \ldots, \pi^{c_d})) \leftarrow$ $d \times d$ matrices such that $A$ is a basis for $\Lambda_2$ and $A\Delta$ is a
    basis for $\Lambda_3$, with $c_1 \leq c_2 \leq \ldots \leq c_d$
for all $i \in \{1, \ldots, d-1\}$ do
    $\lambda \leftarrow c_i$
    $\Gamma_\lambda \leftarrow \mathrm{diag}(\pi^{\max(\lambda, c_1)}, \ldots, \pi^{\max(\lambda, c_d)})$
    $t \leftarrow 0$
    while $\lambda < c_{i+1}$ do
        $\lambda \leftarrow \lambda + t$
        $A_\lambda \leftarrow$ an SA-basis for $(M_1, A\Gamma_\lambda)$
        $(N_1, \ldots, N_d) \leftarrow$ the sequence of base-change matrices in the SA-basis computation
            for $(M_1, A\Gamma_\lambda)$
        $(p_1, \ldots, p_d) \leftarrow$ the pivot position sequence in the SA-basis computation for $(M_1, A\Gamma_\lambda)$

        $L \leftarrow L \cup \{[A_\lambda]\}$
        $t \leftarrow c_{i+1} - c_i$
        for all $j \in \{1, \ldots, d\}$ such that $p_j$ is in the first $i$ columns do
            $v_1 \leftarrow$ valuation of $j$th pivot in $N_j$
            $v_2 \leftarrow$ least valuation among all elements of $N_j$ in columns $i+1, i+2, \ldots, d$ not
                in positions $p_1, \ldots, p_j$
            $t \leftarrow \min(t, v_2 - v_1 + 1)$
        end for
    end while
end for
return $L$
```

Theorem 4.39. *Let $M_1, M_2,$ and M_3 represent three lattices $\Lambda_1, \Lambda_2,$ and Λ_3 in $\mathcal{B}_d$. Then Algorithm 4.38 correctly computes a list L of apartments covering* $\mathrm{conv}(\Lambda_1, \Lambda_2, \Lambda_3)$*, where L has size at most 2^d.*

As before, we can obtain an enveloping membrane for $\Lambda_1, \Lambda_2,$ and Λ_3 by concatenating all matrices in L.

Proof. We retain the setup and variables of Algorithm 4.38. Recall that any class in $\mathrm{conv}(\Lambda_2, \Lambda_3)$ has a representative of the form $\Lambda_3 \cap \pi^\lambda \Lambda_2$. Using A as a basis for Λ_2 and $A\Delta$ as a basis for Λ_3, we see that $\Lambda_3 \cap \pi^\lambda \Lambda_2$ has $A\Gamma_\lambda$ as a basis, where the diagonal matrix $\Gamma_\lambda = \mathrm{diag}(\pi^{\max(\lambda, c_1)}, \ldots, \pi^{\max(\lambda, c_d)})$. Furthermore, if $\lambda < c_1$ then $\Lambda_3 \cap \pi^\lambda \Lambda_2 = \Lambda_3$, and if $\lambda > c_d$ then $\Lambda_3 \cap \pi^\lambda \Lambda_2 = \pi^\lambda \Lambda_2$. Hence we can assume λ is an integer between c_1 and c_d.

It follows from Lemma 4.9 that

$$\mathrm{conv}(\Lambda_1, \Lambda_2, \Lambda_3) = \bigcup_{c_1 \leq \lambda \leq c_d} \mathrm{conv}(M_1, A\Gamma_\lambda).$$

We can therefore cover $\mathrm{conv}(\Lambda_1, \Lambda_2, \Lambda_3)$ with the apartments $[A_\lambda]$ containing $\mathrm{conv}(M_1, A\Gamma_\lambda)$ produced by Algorithm 4.20. By Corollary 4.25, furthermore, if we have computed A_λ already we only need to compute $A_{\lambda+1}$ if some pivot changes position.

Suppose this occurs, with λ in the range $c_i \leq \lambda < c_{i+1}$. Then $A\Gamma_{\lambda+1}$ is obtained from $A\Gamma_\lambda$ by multiplying with the diagonal matrix whose first i diagonal entries are π and last $d-i$ diagonal entries are 1. Let p_j be the earliest pivot which changes positions. By Lemma 4.24, it follows that the jth base-change matrix $N_j^{(\lambda+1)}$ for the pair $(M, A\Gamma_{\lambda+1})$ factors as $N_j^{(\lambda+1)} = N_j^{(\lambda)} \operatorname{diag}(\pi, \ldots, \pi, 1, \ldots, 1)$, where $N_j^{(\lambda)}$ is the jth base-change matrix for the pair $(M, A\Gamma_\lambda)$. Since the jth pivot differs for these two matrices, the jth pivot must appear in the first i columns and there must be an element of equal valuation appearing in the last $d-i$ columns. Conversely, suppose there exists some jth pivot appearing in the first i columns of $N_j^{(\lambda)}$ with an element of equal valuation in the last $d-i$ columns. Then either some earlier pivot already changed, or the jth pivot will be different for $N_j^{(\lambda+1)}$.

It follows that, for λ in the range $c_i \leq \lambda < c_{i+1}$, we can quickly compute the smallest t such that $(M_1, A\Gamma_{\lambda+t})$ will have some jth pivot in a different position than for $(M_1, A\Gamma_\lambda)$. For each jth pivot appearing in the first i columns of $N_j^{(\lambda)}$, we can compare its valuation v_1 to the smallest valuation v_2 of all elements in the last $d-i$ columns of $N_j^{(\lambda)}$. If p_j is the first pivot to change, it will change when $t = t_j := v_2 - v_1 + 1$. So $t = \min(t_j)$ is our desired increment. In particular, Algorithm 4.38 recomputes A_λ each time a pivot changes, so it is indeed correct.

Next we prove that the list L has size at most 2^d. Suppose λ is in the range $c_i \leq \lambda < c_{i+1}$. Our claim is that at most $\binom{d}{i}$ apartments are computed in this range, so that $\sum_i \binom{d}{i} = 2^d$ bounds the number of apartments in L. Write an i-sized subset σ of $[d]$ as $(\sigma_1, \sigma_2, \ldots, \sigma_i)$, where $\sigma_1 < \sigma_2 \cdots < \sigma_i$. We can assign to each λ an i-sized subset σ^λ of $[d]$, where $j \in \sigma^\lambda$ if and only if the jth pivot appears in the first i columns of N_j when computing an SA-basis for M_1 and $A\Gamma_\lambda$. We can also define a well-ordering on the set of all i-sized subsets of $[d]$ lexicographically: $\sigma < \tau$ if and only if the first j with $\sigma_j \neq \tau_j$ satisfies $\sigma_j < \tau_j$. The key insight is that $\sigma^\lambda < \sigma^{\lambda+1}$ if the corresponding pivot sequences for λ and $\lambda+1$ differ. Since there are $\binom{d}{i}$ possible choices for σ^λ, there can be at most $\binom{d}{i}$ different pivot position changes for λ in this range.

It remains to show why this key fact holds. Suppose that incrementing λ by one changes some pivot position, with the jth pivot the first to change. The above analysis shows that the jth pivot for the pair $(M_1, A\Gamma_\lambda)$ must be in the first i columns, and that this must change for the pair $(M_1, A\Gamma_{\lambda+1})$. It follows that j must be in σ^λ, and that j cannot be in $\sigma^{\lambda+1}$. Furthermore, because j is the first pivot to change, for each $\ell < j$ we have $\ell \in \sigma^\lambda \iff \ell \in \sigma^{\lambda+1}$. Hence $\sigma^\lambda < \sigma^{\lambda+1}$, as desired. □

Example 4.40. *Fix $K = \mathbb{Q}_3$ and the building $\mathcal{B}_5$. Let M_1 be the 5×5 identity matrix, and let each entry of M_2 and M_3 be sampled uniformly at random from the finite set $\{3^e : e \in \mathbb{Z}, -20 \leq e \leq 20\}$. The author took 1000 such triangles and computed enveloping membranes via Algorithm 4.38 in Mathematica. After pruning duplicate columns, the matrices describing the enveloping membranes always had at least 6 columns, and at most 25. For comparison, the upper bound implied by our Algorithm 4.38 is $5 \cdot 32 = 160$ columns. A histogram describing the frequency counts for the size of the membranes is presented in Figure 4.9.*

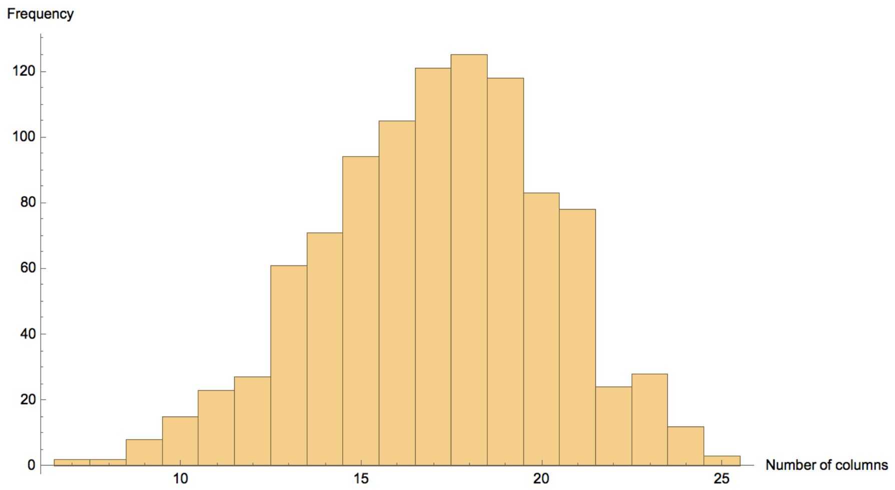


Figure 4.9: Frequency counts for the number of columns of enveloping membranes produced by Algorithm 4.38 for random convex triangles.

One example of a convex triangle attaining the maximal number of columns is given by

$$M_2 = \begin{pmatrix} 3^{-15} & 3^{16} & 3^{-7} & 3^{-8} & 3^{-13} \\ 3^{-13} & 3^{20} & 3^{-12} & 3^{-9} & 3 \\ 3^{-19} & 3^{19} & 3^{7} & 3^{-15} & 3^{10} \\ 3^{9} & 3^{-12} & 3^{-12} & 3^{-17} & 3^{-18} \\ 3^{-17} & 3^{-4} & 3^{-7} & 3^{-3} & 3^{20} \end{pmatrix}, M_3 = \begin{pmatrix} 3^{-1} & 3^{-8} & 3^{-20} & 3^{-1} & 3^{-20} \\ 3^{10} & 3^{6} & 3^{0} & 3^{2} & 3^{-20} \\ 3^{-6} & 3^{8} & 3^{3} & 3^{5} & 3^{-13} \\ 3^{-15} & 3^{9} & 3^{-9} & 3^{2} & 3^{-7} \\ 3^{12} & 3^{-3} & 3^{5} & 3^{-16} & 3^{-13} \end{pmatrix}.$$

After applying Algorithm 4.38 to obtain an appropriate membrane $[M]$, *the author computed the tropical polytope via Algorithm 4.17 presented in Figure 4.10. Note that this convex hull can be spanned by only five of the given points:* $(0, 19, -8)$, $(0, 18, 15)$, $(0, 13, 16)$, $(0, 12, 20)$, *and* $(0, 7, 20)$. *Let* M' *be the square submatrix of* M *with columns corresponding to these five points. Running through Algorithm 4.17 using the apartment* $[M']$ *yields a coarser subdivision of the same tropical polytope. This implies that the convex hull of our three matrices* M_1, M_2, *and* M_3 *all lie in the single common apartment* $[M']$, *which can also be seen using [51, Lemma 25]. That our algorithms do not notice this fact suggests that they likely can be improved, which motivates the following open problem.*

Problem 4.41. *Let* $E(d, s)$ *denote the smallest integer such that any min-convex hull spanned by* s *lattices has an enveloping membrane with representative matrix* M *of size at most* $d \times E(d, s)$. *How does* $E(d, s)$ *vary with* d *and* s*?*

By Remark 4.32, we have that $E(d, s) \leq d \cdot 2^d \cdot (d!)^{s-3} - 1$, *but we expect that this bound is far from tight.*

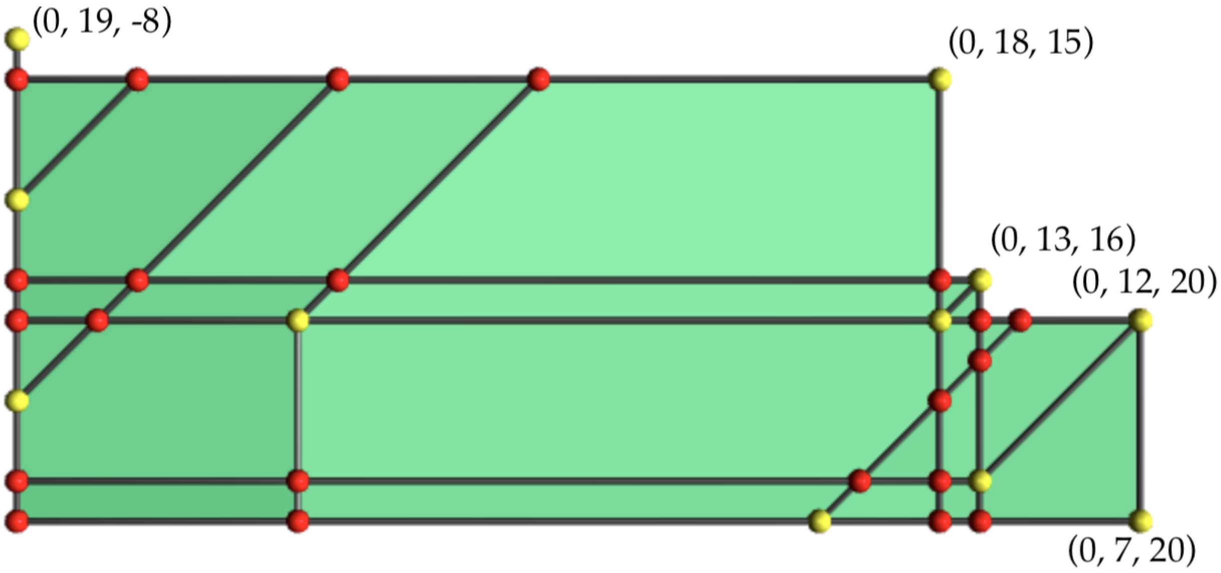


Figure 4.10: The tropical polytope isomorphic to the convex hull of M_1, M_2, M_3, with spanning vertices in yellow. Note that the x- and y-axes have been flipped.

Conclusion

In this chapter, we detailed the first time-bounded complexity methods for the computation of a membrane containing a given min-convex hull in the affine building of SL_d. Our approach was inductive and relied on computing many apartments containing a pair of lattices. As simplicial complexes, the min-convex hull is isomorphic to a tropical polytope and the enveloping membrane to a tropical linear space; our algorithms thus implied bounds on the dimension of the tropical projective space in which these objects are contained. In the following chapter we will see another application of tropical polytopes and linear spaces.

Chapter 5

Tropical principal component analysis

The original material in this chapter is largely based on "Tropical principal component analysis and its application to phylogenetics", joint work with Ruriko Yoshida and Xu Zhang and published in the Bulletin of Mathematical Biology [90]. Section 5.3 originally appeared in the paper "Tropical principal component analysis in the space of phylogenetic trees", joint with Ruriko Yoshida and Robert Page and published in Bioinformatics [89].

5.1 Introduction

Principal component analysis (PCA) is a popular and robust method for reducing the dimension of a high-dimensional data set. Given a positive integer $s \in \mathbb{N}$ and a collection of data points in a high-dimensional Euclidean space $\mathbb{R}^e$, the procedure projects the data points onto a plane of fixed dimension $s-1$, which is obtained by minimizing the sum of squared distances between each point in the dataset and its orthogonal projection onto the plane. This linear plane is a translate of some $(s-1)$-dimensional linear subspace; PCA also constructs an orthonormal basis for that subspace whose vectors are called principal components. The low-dimensional plane is thus described by an $(s \times e)$-dimensional matrix, whose first $s-1$ rows are the principal components and whose last row is the translation vector.

In this chapter we propose two analogous approaches to a principal component analysis in the setting of tropical geometry. Given a positive integer s and a collection of data points in the tropical projective torus, our tropical principal component analyses seek a tropically-geometric object, as close as possible to the data points in the tropical metric d_{tr}. In both cases, furthermore, this tropically-geometric object will be described by an $(s \times e)$-dimensional matrix.

Classically, a full-rank matrix of shape $(s \times e)$ with $s < e$ defines an s-dimensional linear subspace of $\mathbb{R}^e$ via the span of its rows. This subspace is also described by the Plücker coordinates of the matrix. Tropically, on the other hand, these two notions diverge: the tropical Plücker coordinates of a tropical matrix produce a *Stiefel tropical linear space*, defined in [30], while the row-span of the matrix yields a *tropical polytope*. These two notions give rise to our two interpretations of tropical principal component analysis.

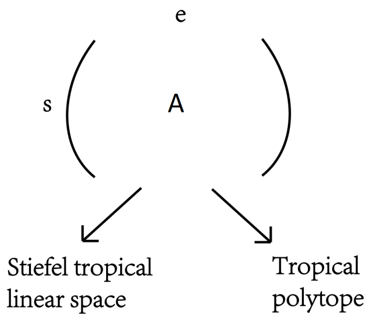


Figure 5.1: A tropical matrix A gives rise to both a Stiefel tropical linear space and a tropical polytope.

We describe our first approach to a tropical principal component analysis in Section 5.2, as the Stiefel tropical linear space closest to the data points under the tropical metric d_{tr}. We call such a Stiefel tropical linear space the *tropical principal linear space*. We give an exact description for a tropical principal hyperplane of e points in terms of the *tropical volume*, originally introduced in [21]. In analogy with [84, Algorithm 4], another method for conducting principal component analysis on the space of phylogenetic trees, we also describe an heuristic algorithm to approximate a best-fit Stiefel tropical linear space of a given dimension.

Next, in Section 5.3 and Section 5.4, we discuss a tropical principal component analysis in terms of best-fit tropical polytopes. We reformulate the problem of finding a best-fit tropical polytope or *tropical principal polytope* in terms of a mixed integer programming problem, then describe an approximative algorithm similar to the above.

We then apply these methods to phylogenetics. The space of rooted equidistant phylogenetic trees with m leaves is naturally embedded into a tropical projective torus as a tropical linear space, so that collections of such trees form a natural tropical dataset. We apply the approximative algorithms for both methods of tropical PCA on a simulated phylogenetic dataset in Section 5.5, and on an empirical dataset of genomes of parasitic protists in the Apicomplex phylum in Section 5.6. In our tropical polytope approach, equidistant trees remain equidistant after projection, and so we examine the distribution of tree topologies in that case.

5.2 Tropical principal linear spaces

As noted in the introduction, one can interpret ordinary $(s-1)$st principal component analysis as a method of dimensionality reduction, replacing data points with their projections onto the translate of some particularly well-fitting linear space of dimension $s-1$. Classically, this translation of a well-fitted linear space can be described by a matrix of size $s \times e$, whose first $(s-1)$ rows are the basis vectors of the linear space, and whose last row is a translation vector from the origin.

In analogy with the classical case, our approach to an $(s-1)$st tropical principal component analysis is to replace data points with their tropical projections onto a best-fit Stiefel tropical linear space of dimension $(s-1)$, defined by a tropical matrix of size $s \times e$, so that the sum of the tropical distances between the data points and their projections is as small as possible. We note that throughout this chapter we seek to minimize the sum of tropical distances, in contrast to the classical case involving the minimization of the sum of squared distances. The ordinary Euclidean distance involves a square root, so that squared distances are often easier to study; since the tropical distance metric is piecewise linear in each coordinate, however, it is natural to simply consider the distances themselves rather than their squares.

Best-fit tropical hyperplanes

We begin our discussion of tropical principal component analysis by considering a specific case: reducing by one the dimension of a collection of e datapoints in $\mathbb{R}^e / \mathbb{R}\mathbf{1}$. In other words, we seek the $(e-1)$st order tropical PCA, or a *best-fit tropical hyperplane*, for a collection of e data points in $\mathbb{R}^e / \mathbb{R}\mathbf{1}$.

We require the following definition, from [21].

Definition 5.1. *Let A be an $e \times e$ matrix with entries in $\mathbb{R}$ whose rows correspond to e points in $\mathbb{R}^e / \mathbb{R}\mathbf{1}$. The* tropical volume *of A is given by the expression*

$$\operatorname{tvol} A := \bigoplus_{\sigma \in S_e} \sum a_{i,\sigma(i)} - \bigoplus_{\tau \in S_e - \sigma_{opt}} \sum a_{i,\tau(i)}, \tag{5.2.1}$$

where σ_{opt} is an optimal permutation attaining the tropical determinant in the first tropical sum.

This quantity can be computed in $O(e^3)$ time [11]. Recall that a square tropical matrix A is *tropically singular* if two distinct permutations attain the tropical determinant. The following, from [74, Lemma 5.1], is one of the earliest results in tropical geometry:

Lemma 5.2. *Let A be an $e \times e$ tropical matrix whose rows represent e points of $\mathbb{R}^e / \mathbb{R}\mathbf{1}$. Then A is tropically singular if and only if those e points lie on a tropical hyperplane in $\mathbb{R}^e / \mathbb{R}\mathbf{1}$. In particular,* $\operatorname{tvol} A = 0$ *if and only if the e points lie on a common tropical hyperplane.*

Of course, if our collection of e datapoints $D^{(i)}$ lie on a common hyperplane, then this hyperplane is our $(e-1)$st tropical PCA. This fact hints at some relationship between the tropical volume and the best fit hyperplane. In fact, this relationship is quite strong.

Theorem 5.3. *Let $D^{(1)}, \dots, D^{(e)}$ be a collection of e points in $\mathbb{R}^e \,/\, \mathbb{R}\,\mathbf{1}$. Then any best-fit hyperplane attains a distance from the e points equal to their tropical volume, and one such best-fit hyperplane is spanned by a choice of $e-1$ of the points.*

To prove this theorem, we first show in the following lemma that the tropical volume is an upper bound on the minimal distance of a best-fit tropical hyperplane.

Lemma 5.4. *Let $D^{(1)}, \dots, D^{(e)}$ be a collection of e points in $\mathbb{R}^e \,/\, \mathbb{R}\,\mathbf{1}$, and let A be the matrix whose i, jth entry is $D^{(i)}_j$. Then there exists a hyperplane of distance $\operatorname{tvol} A$ from the data points, spanned by some choice of $e-1$ of the points.*

Proof. Suppose that all e data points can be spanned by a single hyperplane. Then Lemma 5.2 tells us that this best-fit hyperplane is of distance $\operatorname{tvol} A = 0$ from the data points.

Now suppose that the e data points do not lie on the same hyperplane. Without loss of generality, we may assume that the data points $D^{(1)}, \dots, D^{(e)}$ are ordered so that σ_{opt} in the above definition of the tropical volume is just the identity, and hence the tropical determinant is attained along the diagonal of A.

Let ρ attain the second maximum in the above definition of the tropical volume. Since ρ is not the identity, there must exist some j such that $\rho(j) \neq j$. Let A' be the matrix obtained by deleting the jth row from A, and let p and $\mathcal{H}$ the tropical Plücker vector and tropical hyperplane corresponding to A' as in Example 1.15. The total distance from H to our data points is just the distance from H to $D^{(j)}$, as all other data points are on H by construction.

We compute the difference vector between $D^{(j)}$ and its projection onto H using the Red Rule (Theorem 1.18). The only possible choice for an e-sized subset τ of $[e]$ is just $\tau = [e]$, and we need to compute the maximum and second-maximum values of $p([e]-\tau_i) + D^{(j)}_{\tau_i}$, taken over all choices of $\tau_i \in [e]$. For any such τ_i, we note that $p([e]-\tau_i) + D^{(j)}_{\tau_i}$ is equal to

$$\bigoplus_{\sigma \in S_d,\, \sigma(j)=\tau_i} \sum_i D^{(i)}_{\sigma(i)}.$$

That is, $p_{\tau-\tau_i} + D^{(j)}_{\tau_i}$ is the tropical sum of all permutations which map τ_i to j. In particular, $\tau_i = j$ must yield the largest choice of $p_{\tau-\tau_i} + D^{(j)}_{\tau_i}$, and the second-largest choice must be attained by $\tau_i = \rho^{-1}(j)$. Hence the Red Rule implies that the distance between $D^{(j)}$ and its projection is just the tropical volume, as desired. □

Remark 5.5. *In general, a best-fit Stiefel tropical linear space need not be unique. For example, in the proof of Lemma 5.4, there clearly must be at least two indices j such that $\rho(j) \neq j$.*

We next show that the tropical volume is also a lower bound. To do so, we first derive some intermediate results.

Lemma 5.6. *Let $D^{(1)}, \dots, D^{(e)}$ be a collection of e points in $\mathbb{R}^e / \mathbb{R}\mathbf{1}$, and let A be the $e \times e$ tropical matrix whose i, jth entry is $D^{(i)}_j$. Define the matrix A' whose i, jth entry equals $p([e]-\{i\}) + D^{(j)}_i$, for p any tropical Plücker vector corresponding to a hyperplane. Then A and A' have the same tropical volume.*

Proof. We note that A' is obtained from A by transposition then adding some multiple of $\mathbf{1}$ to each row. Both of these operations preserve the tropical volume of a matrix. □

Now suppose that $\mathcal{H}$ is a tropical hyperplane in $\mathbb{R}^e / \mathbb{R}\mathbf{1}$, and write its corresponding tropical Plücker vector as $p([e]-\{i\})$. We can calculate the distance $\delta_j(\mathcal{H})$ of $\mathcal{H}$ from the jth datapoint $D^{(j)}$ by the Red Rule: the distance is given by

$$\delta_j(\mathcal{H}) = \max_i(p([e]-\{i\}) + D^{(j)}_i) - 2\text{ndmax}_i(p([e]-\{i\}) + D^{(j)}_i),$$

where $2\text{ndmax}_i(p([e]-\{i\}) + D^{(j)}_i) = \max_{i \in [n]-\{i_{opt}\}}(p([e]-\{i\}) + D^{(j)}_i)$, and i_{opt} is some index that attains the maximum in $\max_i(p([e]-\{i\}) + D^{(j)}_i)$.

We write the total distance of $\mathcal{H}$ from our datapoints as $d(\mathcal{H})$. It is given by

$$\begin{aligned} &d(\mathcal{H}) = \textstyle\sum_j \delta_j(\mathcal{H}) \\ &= \textstyle\sum_j \left(\max_i(p([e]-\{i\}) + D^{(j)}_i) - 2\text{ndmax}_i(p([e]-\{i\}) + D^{(j)}_i)\right). \end{aligned}$$

We can rewrite the cost function $d(\mathcal{H})$ above by grouping together the summed and subtracted terms. For fixed j, define $\alpha_j(\mathcal{H}) = \max_i(p([e]-\{i\}) + D^{(j)}_i)$ and $\beta_j(\mathcal{H}) = 2\text{ndmax}_i(p([e]-\{i\}) + D^{(j)}_i)$. Then $\delta_j(\mathcal{H}) = \alpha_j(\mathcal{H}) - \beta_j(\mathcal{H})$, and the cost function can also be written as

$$d(\mathcal{H}) = \sum_j \delta_j(\mathcal{H}) = \sum_j \alpha_j(\mathcal{H}) - \sum_j \beta_j(\mathcal{H}).$$

Definition 5.7. *Fix j in the cost function above, and let i_1 and i_2 be distinct indices such that $\alpha_j(\mathcal{H}) = p([e]-\{i_1\}) + D^{(j)}_{i_1}$ and $\beta_j(\mathcal{H}) = p([e]-\{i_2\}) + D^{(j)}_{i_2}$. If $\delta_j(\mathcal{H}) = 0$, meaning that $\alpha_j(\mathcal{H}) = \beta_j(\mathcal{H})$, we say that the two indices i_1 and i_2* appear in a tie *for index j. If there exists another index i_3 such that $p([e]-\{i_3\}) + D^{(j)}_{i_3} = \alpha_j(\mathcal{H}) = \beta_j(\mathcal{H})$, we call this a* multiple tie *for index j; if there does not exist such an i_3, we call this a* two-way tie.

Note that, in the event of a tie, we may choose any two of the indices attaining the tie to correspond to $\alpha_j(\mathcal{H})$ and $\beta_j(\mathcal{H})$.

Lemma 5.8. *Let $\mathcal{H}$ be an optimal hyperplane in $\mathbb{R}^e / \mathbb{R}\mathbf{1}$, and let p be its corresponding tropical Plücker vector. Choose an index i such that $p([e]-\{i\}) < \beta_j \le \alpha_j$ for all j. Then we can perturb $\mathcal{H}$ to obtain a new best-fit hyperplane $\mathcal{H}'$ so that $p([e]-\{i\}) + D^{(j)}_i = \beta_j$ for some j, and this j corresponds to a multiple tie.*

Proof. Because $p([e]-\{i\})$ does not appear in the cost function by assumption, by Remark 1.21 we can find a new hyperplane $\mathcal{H}'$ with the same tropical Plücker vector as $\mathcal{H}$ except for a larger value for $p([e]-\{i\})$.

If we make $p([e]-\{i\})$ large enough, it must appear in the cost function for $\mathcal{H}'$. In fact, it must appear as part of a multiple tie. If it were a second maximum not equal to the maximum, then $\mathcal{H}'$ would be a better-fitting hyperplane. □

Lemma 5.9. *Let A be an $e \times e$ matrix with entries in $\mathbb{R}$ whose rows correspond to points in $\mathbb{R}^e / \mathbb{R}\mathbf{1}$, and let A' be constructed from A as in Lemma 5.6. Then the tropical volume of A is a lower bound for the cost function. Furthermore, we have that $\sum \alpha_j = \operatorname{tdet} A'$.*

Proof. Let $\mathcal{H}$ be a best-fit hyperplane in $\mathbb{R}^e / \mathbb{R}\mathbf{1}$ for the rows of A, with corresponding tropical Plücker vector p. The basic argument is as follows: we can perturb $\mathcal{H}$ to obtain a new best-fit hyperplane whose sum of distances to the data points given by the Red Rule is the difference of two permutations, with the larger permutation corresponding to the tropical determinant of A'.

We prove the result by induction on e. For the base case, let $e = 1$. Then the tropical volume and the cost function are both trivial.

Suppose we have proved the lemma up to $e-1$. We divide the situation into several possible cases. First, let there be some index k appearing only in ties in the cost function, with at most one of these appearances being a two-way tie. If k appears in a two-way tie, let $D^{(j_k)}$ denote the corresponding datapoint. Otherwise, let $D^{(j_k)}$ denote some datapoint for which k appears in a multiple tie.

Then we can write total distance as

$$p([e]-\{k\}) + D_k^{(j_k)} - p([e]-\{k\}) - D_k^{(j_k)} + \sum_{j \neq j_k} \delta_j(\mathcal{H}).$$

Construct the matrix A'' by deleting the kth row and j_kth column from A'. We also define the hyperplane $\mathcal{H}' \subseteq \mathbb{R}^{e-1} / \mathbb{R}\mathbf{1}$ obtained by “deleting” the index $\{k\}$ from $[e]$: the tropical Plücker vector p' corresponding to $\mathcal{H}'$ is defined by

$$p'([e-1]-\{i\}) = \begin{cases} p([e]-\{i\}) & \text{if } i < k \\ p([e]-\{i+1\}) & \text{if } i \geq k \end{cases}.$$

Because we assumed that k appears in at most one two-way tie, for any $j \neq j_k$ we can choose the indices corresponding to $\alpha_j(\mathcal{H})$ and $\beta_j(\mathcal{H})$ so that k does not appear in $\alpha_j(\mathcal{H}) - \beta_j(\mathcal{H}) = \delta_j(\mathcal{H})$. By construction, therefore, $d(\mathcal{H}) = \sum_{j \neq j_k} \delta_j(\mathcal{H})$ is also the distance between $\mathcal{H}' \subseteq \mathbb{R}^{e-1} / \mathbb{R}\mathbf{1}$ and the rows of the matrix A''. Furthermore, the optimality of $\mathcal{H}$ implies that $\mathcal{H}'$ must be a best-fit tropical hyperplane for the rows of A''.

In particular, the inductive hypothesis states that $d(\mathcal{H}') = \sum_{j \neq j_k} \delta_j(\mathcal{H})$ is bounded from below by the tropical volume of A''. It also implies that $\sum_{j \neq j'} \alpha_j(\mathcal{H}) = \operatorname{tdet} A''$. It therefore follows that $d(\mathcal{H}) = d(\mathcal{H}')$ is bounded below by a difference of distinct permutations in A', and that $\sum \alpha_j(\mathcal{H})$ equals a sum of terms of A' corresponding to some permutation of S_e.

In fact, since each α_j is the largest term in the jth row of A', we must have that $\sum \alpha_j = \operatorname{tdet} A'$. Hence we have for some $\sigma \in S_e$,

$$d(\mathcal{H}) \geq \operatorname{tdet} A' - \sum_i a_{i,\sigma(i)} \geq \operatorname{tvol} A' = \operatorname{tvol} A$$

where the last equality holds by Lemma 5.6.

Now suppose that there exists an index k such that $p([e] - \{k\})$ does not appear in any terms in the cost function. Then by Lemma 5.8, we may replace $\mathcal{H}$ with another hyperplane such that k appears only in a multiple tie for some index j. We are now in the previous case, and the same argument holds as before.

Finally, suppose that for each index i, either $p([e] - \{i\})$ appears in the cost function as part of a non-tie, or $p([e] - \{i\})$ appears in at least two two-way ties. Pick i_1 such that $\alpha_j(\mathcal{H}) = p([e] - \{i_1\}) + D_{i_1}^{(j)}$ for j corresponding to a non-tie. We write this index j as j_{i_1}, and we write i_0 as the index corresponding to $\beta_{i_{j_1}}(\mathcal{H})$. Suppose that there does not exist some other index j_{i_2} such that $\beta_{j_{i_2}} = p([e] - \{i_1\}) + D_{i_2}^{(j_{i_2})}$. Then we could perturb $\mathcal{H}$ by slightly lowering $p_{[e]-\{i_1\}}$ to obtain a better-fitting hyperplane, a contradiction. Hence such a j_{i_2} must exist.

In fact, note that we can pick j_{i_2} to avoid a multiple-way tie at that index. Otherwise, perturbing $p([e] - \{i_1\})$ upward would not affect the second and first minimum, and we could obtain the same contradiction. It follows that the index j_{i_2} must correspond to either a two-way tie or a non-tie. In either case, therefore, there is a unique other index i_2 such that $\alpha_{j_{i_2}} = p([e] - \{i_2\}) + D_{i_2}^{(j_{i_2})}$.

If the cost function term corresponding to j_{i_2} is a non-tie, and i_2 appeared in no other cost function terms as part of the subtracted term, then we can obtain a contradiction in a similar way as above by perturbing $p([e] - \{i_2\})$. If the cost function term corresponding to j_{k_2} is a tie, and i_1 and i_2 appeared in no other cost function terms as part of the subtracted term, then we could obtain a contradiction in a similar way as above by perturbing $p([e] - \{i_1\})$ and $p([e] - \{i_2\})$ in sync.

Hence in a similar fashion we may obtain indices i_3, and a i_4, and so on, such that each $i_k = \alpha_{j_{i_k}}(\mathcal{H})$ for some index j_{i_k} corresponding to either a two-way tie or a non-tie. Because there can only be at most e such indices j_{i_k}, there must exist ℓ and ℓ' such that $i_\ell = i_{\ell'}$ with $\ell > \ell'$. If $\ell' \neq 0$, then we may repeat the argument by perturbing $p([e] - \{i_{\ell'-1}\})$ upward, possibly in tandem with some earlier Plücker coordinates. Hence we must find $i_\ell = i_0$ for some ℓ.

If $\ell < e$, and if there exists another index $i_{\ell+1}$ which appears as a positive term in the cost function, we repeat the above argument. It therefore follows that if $p([e] - \{i\})$ appears in the cost function as part of a non-tie, it must appear at least twice as part of a non-tie or a two-way tie. By assumption, therefore, each index appears at least twice as part of a non-tie or a two-way tie.

In particular, the pigeonhole principle implies that each index i appears exactly twice as part of a non-tie or a two-way tie. It can thus be assumed that each index appears once as part of some α_i and once as part of some β_i. Now the distance function $d(\mathcal{H})$ is the difference between two different permutations of S_e. As before, $\sum_{\alpha_i}$ must therefore equal the tropical determinant of A', and the distance function $d(\mathcal{H})$ must be bounded below by the tropical volume as desired. □

Together, Lemmas 5.4 and 5.9 imply Theorem 5.3. This result provides a new interpretation for the tropical volume of a collection of e points: it measures the deviation of those points from lying on a common hyperplane. It also suggests a possible extension of the definition of a tropical volume to rectangular matrices ([21, Section 5]): the tropical volume of a "skinny" matrix with more rows than columns could be defined as the sum of the distances of the row-points from a best-fit tropical hyperplane.

If $n > e$, an optimist might hope that the best-fit tropical hyperplane of n points in $\mathbb{R}^e / \mathbb{R}\mathbf{1}$ would again attain a total distance equal to the tropical volume of some subset of e of those points. In fact, this does not hold even for $e = 3$:

Example 5.10. *Consider the matrix A whose rows correspond to data points in $\mathbb{R}^3 / \mathbb{R}\mathbf{1}$:*

$$A = \begin{pmatrix} 0 & -2 & -2 \\ 0 & -1 & 2 \\ 0 & 2 & -1 \\ 0 & 2 & 2 \end{pmatrix}.$$

The tropical volume of the first three points in A is 4, so any tropical line must attain a distance at least 4 to the four points. This is attained by the tropical line with apex at $(0, 2, 2)$.

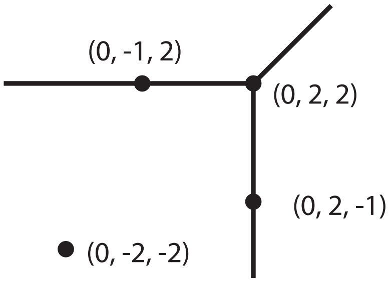


Figure 5.2: A best-fit tropical line for the data in Example 5.10.

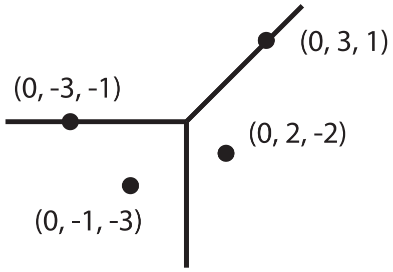


Figure 5.3: A best-fit tropical line for the data in Example 5.11.

Example 5.11. *Let A be the following matrix whose rows correspond to data points in $\mathbb{R}^3 / \mathbb{R}\mathbf{1}$:*

$$A = \begin{pmatrix} 0 & -1 & -3 \\ 0 & 2 & -2 \\ 0 & 3 & 1 \\ 0 & -3 & -1 \end{pmatrix}.$$

The largest tropical volume of any choice of three rows is 2, but inspection shows that a best-fit tropical line attains a total distance of 3.

Best-fit Stiefel tropical linear spaces

In view of Theorem 5.3 and Lemma 5.4, we describe an algorithm to approximate a best-fit Stiefel tropical linear space of any given dimension. Our method is very similar to [84, Algorithm 4]; Algorithm 5.24 below also follows the same general approach. For simplicity, below we state the algorithm for a Stiefel tropical linear space of dimension 2.

Algorithm 5.12 (Stochastic approximation of a second-order tropical principal subspace)**.**

```
Require: A dataset $D^{(i)}$ of points in the tropical projective torus
Ensure: A Stiefel tropical linear space $L_p$ close to $D^{(i)}$
Fix an ordered set $V = (D^{(1)}, D^{(2)}, D^{(3)})$ and compute $L_p(V)$.
repeat
    Sample three datapoints $D^{(j_1)}, D^{(j_2)}, D^{(j_3)}$ randomly from the set of all datapoints.
    Let $V' = \{D^{(j_1)}, D^{(j_2)}, D^{(j_3)}\}$.
    Compute $d(L_p(V'))$.
    if $d(L_p(V)) > d(L_p(V'))$, set $V \leftarrow V'$.
until convergence.
```

This algorithm attempts to minimize $d(L_p)$ by randomly varying the three points generating L_p within the set of all datapoints. Whenever a choice of three points V' improves upon the current configuration V, we replace V with V'. Convergence is assessed by considering whether a new choice of V has been found over a fixed number of previous iterations; if no better V is found over some prespecified number of iterations, then the algorithm terminates. We note that under the conditions of Theorem 5.3, Algorithm 5.12 is very likely to construct an exact best-fit Stiefel tropical linear space as described in Lemma 5.4; in this sense the algorithm attempts to generalize the results of Theorem 5.3.

Estimating the empirical principal components was first considered in [29]. Feragen et. al estimated the first order principal component by spanning two well-chosen data points in the BHV space.

Algorithm 5.12 is probabilistic in nature, so we cannot provide deterministic complexity bounds. Because the Red and Blue Rules for calculating projections onto L_p are relatively expensive to compute naively, Algorithm 5.12 may be slow to execute.

Remark 5.13. *Algorithm 5.12 does not always attain an exact best-fit Stiefel tropical linear space. This is clear, for example, if we consider a variant of the algorithm for fitting a 0-dimensional Stiefel tropical linear space, i.e., a tropical Fermat-Weber point as in [62]. In general, the collection of tropical Fermat-Weber points for a given dataset need not contain a data point.*

Because the space of ultrametrics $\mathcal{U}_m$ is a tropical linear space (Theorem 1.29), which is tropically convex, the convex hull of points in $\mathcal{U}_m$ is contained in $\mathcal{U}_m$. Unfortunately, however, the Stiefel tropical linear space defined by these points may not itself be contained in $\mathcal{U}_m$.

Lemma 5.14. *Let L_p be a tropical linear space and $D^{(i)} \in L_p$ some points in the tropical linear space. It need not be the case that the Stiefel tropical linear space L_q defined by the points is contained in L_p.*

Proof. For a very simple counterexample, let L_p be the tropical line in $\mathbb{R}^3 / \mathbb{R}\mathbf{1}$ centered at the origin, and take the two points $D^{(1)} = (0, -1, 0)$ and $D^{(2)} = (0, -2, 0)$. We have the picture in Figure 5.4. □

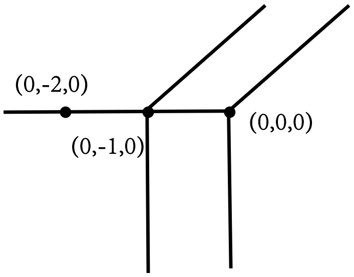


Figure 5.4: Both $(0, -2, 0)$ and $(0, -1, 0)$ are contained in the Stiefel tropical linear space spanned by $(0, -2, 0)$ and $(0, 0, 0)$, but the Stiefel tropical linear space spanned by the two points is not.

If our data points $D^{(i)}$ correspond to ultrametrics, by Lemma 5.14 the Stiefel tropical linear space produced by Algorithm 5.12 may not be contained in the overall space of ultrametrics. Hence this approach does not apply directly to the analysis of equidistant trees.

In the proof of Lemma 5.14, however, if our two chosen points $D^{(1)}$ and $D^{(2)}$ lie on different rays of the tropical line L_p, it is easy to see that their corresponding Stiefel tropical linear space will be contained in L_p as well. In general, given some points $D^{(i)}$ in a tropical linear space L_p, it would be interesting to study the conditions under which their corresponding Stiefel tropical linear space L_q satisfies $L_q \subseteq L_p$. Such a result would enable a natural extension of these methods to the study of ultrametrics.

The classical principal components have a nested structure, in which the ith principal subspace is contained in the $(i + 1)$st subspace for each i. It is natural to wonder whether a similar relationship holds in this tropical analogue. Again, the situation is complicated.

Example 5.15. *The minimal distance of a zeroth tropical principal component, or a* tropical Fermat-Weber point*, is given in [62, Theorem 3].*

Let $D^{(1)} = (0, -2, -2), D^{(2)} = (0, -1, 2)$, and $D^{(3)} = (0, 2, -1)$. Then their tropical volume equals 4, and a tropical Fermat-Weber point attains a total distance of seven from the three points. A best-fit hyperplane is given by the line with apex at $(0, 1, -2)$, and inspection shows that no point on this line is a Fermat-Weber point.

On the other hand, the point $(0, -1, -1)$ can be checked to be a Fermat-Weber point, and the line with apex at $(0, 2, -1)$ is a best-fit hyperplane containing that Fermat-Weber point. In other words, a best-fit tropical line need not contain a best-fit tropical point, but we can find an example in this case for which this containment holds.

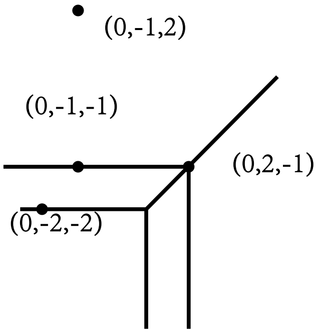


Figure 5.5: Both tropical lines attain a minimum sum of distances from the points $(0, -1, 2)$, $(0, -2, -2)$, and $(0, 2, -1)$. But only one contains the Fermat-Weber point $(0, -1, -1)$.

5.3 Tropical principal polytopes

We now discuss a different notion of a tropical principal component analysis, in which our analogue to a linear plane is a tropical polytope called a *tropical principal polytope.* Classically, the row-span of a matrix of dimensions $s \times e$ defines a linear space of dimension at most s. In the tropical setting, by contrast, Section 1.3 tells us that the row-span of a tropical matrix is a tropical polytope.

A tropical principal component analysis, therefore, outputs the tropical convex hull of s points in $\mathbb{R}^e / \mathbb{R}\,\mathbf{1}$ minimizing the sum of distances between each point in the sample to its projection onto the convex hull.

Definition 5.16. *Let $\mathcal{P} = \text{tconv}(D^{(1)}, \ldots, D^{(t)}) \subseteq \mathbb{R}^e / \mathbb{R}\,\mathbf{1}$ be a tropical polytope with vertices $\{D^{(1)}, \ldots, D^{(t)}\} \subset \mathbb{R}^e/\mathbb{R}\mathbf{1}$ for $1 \leq t \leq (e-1)$ and let $S = \{u_1, \ldots u_n\}$ be a sample from the space of ultrametrics $\mathcal{U}_m$. Let $\Pi_{\mathcal{P}}(S) := \sum_{i=1}^{n} d_{tr}(u_i, u_i')$, where u_i' is the tropical projection of u_i onto the tropical polytope $\mathcal{P}$. Then $\mathcal{P}$ is called the $(t-1)$st order tropical principal polytope of S and the vertices $D^{(1)}, \ldots, D^{(t)}$ of the tropical polytope $\mathcal{P}$ are called the $(t-1)$st* *order tropical principal components of S if the tropical polytope $\mathcal{P}$ minimizes $\Pi_{\mathcal{P}}(S)$ over all possible tropical polytopes with t many vertices.*

In comparing tropical PCA with classical PCA, we remark that the tropical metric in tropical projective space is closely related to the L_∞ norm (for example, [5, 68]). The tropical norm of $x \in \mathbb{R}^e/\mathbb{R}\mathbf{1}$ can be seen as the L_∞-norm of the representative of x with minimal coordinate 0. Also, it is twice the quotient norm of the L_∞ norm [41, Lemma 5.2]. In this sense, classical PCA is built on L_2 minimization while tropical PCA is built on L_∞ minimization.

It is important to be able to interpret the tropical principal components of a dataset of equidistant trees in terms of phylogenetics. To that end, we seek to prove properties about its interpretation. One nice property of tropical principal polytopes is that each cell comprises ultrametrics of the same tree topology., as is visible in e.g. Figure 5.10.

Theorem 5.17. *Let $\mathcal{P} = \mathrm{tconv}(D^{(1)}, \ldots, D^{(t)}) \subseteq \mathbb{R}^e/\,\mathbb{R}\,\mathbf{1}$ be a tropical polytope spanned by ultrametrics. Then any two points x and y in the same cell of $\mathcal{P}$ are also ultrametrics with the same tree topology.*

Proof. The space of ultrametrics $\mathcal{U}_m$ is a tropical linear space, so it is tropically convex, and $\mathcal{P}$ is contained in $\mathcal{U}_m$. Hence any points x and y in $\mathcal{P}$ must also be ultrametrics.

Let Q be the type of x and y. To check whether x and y have the same tree topology, we check the three point condition for each trio of leaves. Fix such a trio i, j, and k. Our first claim is that $x_{ij} = x_{ik} = x_{jk}$ if and only if $y_{ij} = y_{ik} = y_{jk}$. To see why, suppose the former is true. Because $x \in \mathcal{P}$, there exists some index $a \in Q_{ij}$, so that

$$D_{ij}^{(a)} - x_{ij} = \max_{\ell_1 < \ell_2} D_{\ell_1,\ell_2}^{(a)} - x_{\ell_1,\ell_2}.$$

In particular, $D_{ij}^{(a)} - x_{ij} \geq D_{ik}^{(a)} - x_{ik}$ and $D_{ij}^{(a)} - x_{ij} \geq D_{jk}^{(a)} - x_{jk}$. By assumption it follows that $D_{ij}^{(a)} \geq D_{ik}^{(a)}$ and $D_{ij}^{(a)} \geq D_{ik}^{(a)}$. Because $D^{(a)}$ is an ultrametric, the maximum among $D_{ij}^{(a)}, D_{ik}^{(a)}, D_{jk}^{(a)}$ is attained twice, so one of these is actually an equality. Without loss of generality, let $D_{ij}^{(a)} = D_{ik}^{(a)}$. Then $D_{ij}^{(a)} - x_{ij} = D_{ik}^{(a)} - x_{ik}$ and $a \in Q_{ik}$ as well.

Recall that Q is also the type of y. This means

$$D_{ij}^{(a)} - y_{ij} = D_{ik}^{(a)} - y_{ik} = \max_{\ell_1 < \ell_2} D_{\ell_1,\ell_2}^{(a)} - y_{\ell_1,\ell_2}.$$

In particular, since $D_{ij}^{(a)} = D_{ik}^{(a)}$, we have $y_{ij} = y_{ik}$ as well. The same argument applied to Q_{jk} shows that $y_{jk} = y_{ij}$ or $y_{jk} = y_{ik}$; it follows that $y_{ij} = y_{ik} = y_{jk}$ as desired.

Now suppose $\max(x_{ij}, x_{ik}, x_{jk})$ and $\max(y_{ij}, y_{ik}, y_{jk})$ are both attained exactly twice. We claim that the minimum for x and for y is attained for the same pair of leaves. Suppose without loss of generality that $x_{ij} = \min(x_{ij}, x_{ik}, x_{jk})$ and $y_{ik} = \min(y_{ij}, y_{ik}, y_{jk})$: because $x, y \in \mathcal{P}$, there exists some index $a \in S_{jk}$. This implies in particular that $D_{jk}^{(a)} - x_{jk} \geq D_{ij}^{(a)} - x_{ij}$ and $D_{jk}^{(a)} - y_{jk} \geq D_{ik}^{(a)} - y_{ik}$.

Rearranging these inequalities produces $D^{(a)}_{jk} - D^{(a)}_{ij} \geq x_{jk} - x_{ij}$ and $D^{(a)}_{jk} - D^{(a)}_{ik} \geq y_{jk} - y_{ik}$. Since $x_{ij} < x_{jk}$ and $y_{ik} < y_{jk}$ by assumption, it follows that $D^{(a)}_{jk}$ must be the unique maximum among $D^{(a)}_{ij}, D^{(a)}_{ik}, D^{(a)}_{jk}$, a contradiction because $D^{(a)}$ is ultrametric. □

It is natural to ask when a $(t-1)$st order tropical principal polytope contains the origin $\mathbf{0}$: i.e., when the fully unresolved phylogenetic tree (the star tree) is contained in the $(t-1)$st order tropical principal polytope. The origin $\mathbf{0}$ is also the star tree in the BHV metric. This question turns out to have a simple answer.

Lemma 5.18. *Let $\mathcal{P} = \operatorname{tconv}(D^{(1)}, \ldots, D^{(t)}) \subseteq \mathbb{R}^e/\mathbb{R}\mathbf{1}$ be a tropical polytope spanned by ultrametrics. The origin $\mathbf{0}$ is contained in $\mathcal{P}$ if and only if the path between each pair of leaves i, j passes through the root of $D^{(k)}$ for some $k \in [t]$.*

Proof. The vertices $D^{(1)}, \ldots, D^{(t)}$ can certainly be tropically scaled to have largest coordinate 0. If the claimed condition holds, then the sum of these scaled $D^{(i)}$ will be the origin as desired.

Suppose $\mathbf{0}$ is contained in $\mathcal{P}$, so we can write $\mathbf{0} = \bigoplus a_k \odot D^{(k)}$. Consider some pair of leaves i, j. We know that 0 must appear as the (i,j)-coordinate of some particular $a_l \odot D^{(l)}$. Because $\mathbf{0} = \bigoplus a_k \odot D^{(k)}$, all other coordinates of $a_l \odot D^{(l)}$ must be non-positive, meaning that the (i,j)-coordinate of $a_l \odot D^{(l)}$ is maximal as desired. □ □

Definition 5.19 ([62]). *Suppose we have a sample $\{D^{(1)}, \ldots, D^{(n)}\}$. A* Fermat-Weber point *x^* of $\{D^{(1)}, \ldots, D^{(n)}\}$ is a minimizer of the sum of tropical distances to the data points:*

$$x^* := \operatorname{argmin}_x \sum_{i=1}^{n} d_{\mathrm{tr}}(x, D^{(i)}).$$

We can naturally view Fermat-Weber points as zero-dimensional tropical principal polytopes.

Lemma 5.20. *Suppose $n \geq 3$ and $\{D^{(1)}, \ldots, D^{(n)}\} \subset \mathcal{U}_m$. Then there exists a Fermat-Weber point x^* of the dataset lying in the tropical polytope $\operatorname{tconv}(D^{(1)}, \ldots, D^{(n)})$. In particular, this point x^* is ultrametric.*

Proof. Take x^* a Fermat-Weber point not lying in the polytope $\operatorname{tconv}(D^{(1)}, \ldots, D^{(n)})$, and let $Q = (Q_1, \ldots, Q_e)$ be its vector of types as in Section 1.3. Because x^* does not lie in the tropical polytope, some of the types Q_j are empty. Consider such a Q_j. By definition, we have that for each i, $D^{(i)}_j - x^*_j$ is not maximal among $\{D^{(i)}_1 - x^*_1, \ldots, D^{(i)}_e - x^*_e\}$. This also means that we can shift the jth coordinate without changing the distance of x^* to any datapoint $D^{(i)}$. We can therefore simply decrease x^*_j until there is some i such that $D^{(i)}_j - x^*_j$ is tied for being maximal among the coordinates of $D^{(i)} - x^*$. The tropical type Q_j of our new x^* will be nonempty. By doing so for all coordinates, we obtain a new Fermat-Weber point which lies in the tropical polytope spanned by ultrametrics and so is itself ultrametric. □

The previous lemma states that there always exists a biologically interpretable zero-dimensional tropical principal polytope for a dataset of ultrametrics. This result points toward the following conjecture, which is analogous to the classical fact that the $(t-1)$st order principal component subspace is contained in the $(t'-1)$st order principal component subspace if $t \leq t'$.

Conjecture 5.21. *There exists a tropical Fermat-Weber point $x^* \in \mathcal{U}_m$ of a sample of ultrametric tree datapoints $D^{(1)}, \ldots, D^{(n)}$ which is contained in the $(t-1)$st order tropical principal polytope of the dataset for $1 \leq t \leq (e-1)$.*

5.4 Computing tropical principal polytopes

We now describe a reformulation of the tropical principal polytope as an optimization problem. For simplicity of exposition, we focus on the second order principal components, noting that the following discussion could be generalized to arbitrary s.

Proposition 5.22. *Solving for the second-order tropical principal polytope can be formulated as the following optimization problem:*

$$\begin{aligned}
\textit{minimize} \quad & \sum_{i=1}^{n} \Delta_i && && (5.4.1)\\
\textit{subject to:} \quad & \Delta_i \geq d_i(k) - d_i^{'}(k) - d_i(l) + d_i^{'}(l), && 1 \leq k < l \leq e \\
& \Delta_i \geq -[d_i(k) - d_i^{'}(k) - d_i(l) + d_i^{'}(l)], && 1 \leq k < l \leq e \\
& d_i^{'}(k) - (\lambda_p^i + D^{(p)}(k)) \geq 0, \\
& d_i^{'}(k) - (\lambda_p^i + D^{(p)}(k)) \leq u_{pik} \times y_{pik}, \\
& \sum_{p=1}^{3} y_{pik} \leq 2, \\
& 0 \leq y_{pik} \leq 1, y_{pik} \textit{ is an integer}, \\
& d_i(k) - (\lambda_p^i + D^{(p)}(k)) \geq 0, \\
& d_i(k) - (\lambda_p^i + D^{(p)}(k)) \leq v_{pik} \times z_{pik}, \\
& \sum_{k=1}^{e} z_{pik} \leq e-1, \\
& 0 \leq z_{pik} \leq 1, z_{pik} \textit{ is an integer}, \\
& \forall p \in [3], k \in [e], i \in [n]
\end{aligned}$$

where u_{pik} and v_{kip} are large enough constants.

Proof. Our optimization problem can be written more explicitly as

$$\min_{D^{(1)},D^{(2)},D^{(3)}\in\mathbb{R}^e/\mathbb{R}\mathbf{1}} \sum_{i=1}^{n} \max\{|d_i(k)-d_i^{'}(k)-d_i(l)+d_i^{'}(l)| : 1\le k<l\le e\}$$

where $d_i^{'} = \lambda_1^i \odot D^{(1)} \oplus \lambda_2^i \odot D^{(2)} \oplus \lambda_3^i \odot D^{(3)},$ with $\lambda_k^i = \min(d_i - D^{(k)})$ and $k = 1,2,3$. Define the quantity

$$\Delta_i = \max\{|d_i(k)-d_i^{'}(k)-d_i(l)+d_i^{'}(l)| : 1\le k<l\le e\}, i\in[n].$$

Then the objective function is equivalent to

$$\begin{aligned} \text{minimize: } & \sum_{i=1}^{n}\Delta_i \\ \text{subject to: } & \Delta_i \ge |d_i(k)-d_i^{'}(k)-d_i(l)+d_i^{'}(l)|, && 1\le k<l\le e. \end{aligned}$$

These constraints can be reformulated as:

$$\begin{aligned} \text{subject to: } & \Delta_i \ge d_i(k)-d_i^{'}(k)-d_i(l)+d_i^{'}(l), && 1\le k<l\le e \\ & \Delta_i \ge -[d_i(k)-d_i^{'}(k)-d_i(l)+d_i^{'}(l)], && 1\le k<l\le e. \end{aligned}$$

Recall the definitions

$$d_i^{'}(k) = \max(\lambda_1^i + D^{(1)}(k), \lambda_2^i + D^{(2)}(k), \lambda_3^i + D^{(3)}(k)),$$

where $\lambda_s^i = \min(d_i - D^{(s)})$. These are equivalent to

$$\begin{aligned} d_i^{'}(k) = \text{ maximize: } & \lambda_1^i + D^{(1)}(k), \lambda_2^i + D^{(2)}(k), \lambda_3^i + D^{(3)}(k), \\ \text{subject to: } & \lambda_1^i \le d_i(t) - D^{(1)}(t), && t\in[e] \\ & \lambda_2^i \le d_i(t) - D^{(2)}(t), && t\in[e] \\ & \lambda_3^i \le d_i(t) - D^{(3)}(t), && t\in[e]. \end{aligned}$$

We can hence divide our original maximization problem into two parts:
for all $k = 1,2,3,\ldots,e$,

$$\begin{aligned} d_i^{'}(k) = \text{ maximize: } & \lambda_1^i + D^{(1)}(k), \lambda_2^i + D^{(2)}(k), \lambda_3^i + D^{(3)}(k), \\ \text{subject to: } & \lambda_1^i \le d_i(t) - D^{(1)}(t), && t\in[e] \\ & \lambda_2^i \le d_i(t) - D^{(2)}(t), && t\in[e] \\ & \lambda_3^i \le d_i(t) - D^{(3)}(t), && t\in[e]. \end{aligned}$$

and

$$\begin{aligned} \text{minimize: } & \sum_{i=1}^{n} \Delta_i & \\ \text{subject to: } & \Delta_i \geq d_i(k) - d_i^{'}(k) - d_i(l) + d_i^{'}(l), & 1 \leq k < l \leq e \\ & \Delta_i \geq -[d_i(k) - d_i^{'}(k) - d_i(l) + d_i^{'}(l)], & 1 \leq k < l \leq e. \end{aligned}$$

We recombine them into one optimization as follows:

$$\begin{aligned} \text{minimize:} & \sum_{i=1}^{n} \Delta_i & \\ \text{subject to: } & \Delta_i \geq d_i(k) - d_i^{'}(k) - d_i(l) + d_i^{'}(l), & 1 \leq k < l \leq e, i \in [n] \\ & \Delta_i \geq -[d_i(k) - d_i^{'}(k) - d_i(l) + d_i^{'}(l)], & 1 \leq k < l \leq e, i \in [n] \\ & d_i^{'}(k) \geq \lambda_p^i + D^{(p)}(k), & p = 1,2,3, k \in [e], i \in [n] \\ & \prod_{p=1}^{3} [d_i^{'}(k) - (\lambda_p^i + D^{(p)}(k))] = 0, & k \in [e], i \in [n] \\ & \lambda_p^i + D^{(p)}(t) \leq d_i(t), & p = 1,2,3, t \in [e], i \in [n] \\ & \prod_{t=1}^{e} [d_i(t) - (\lambda_p^i + D^{(p)}(t))] = 0 & p = 1,2,3, i \in [n]. \end{aligned}$$

The result now follows by adding new binary variables y_{pik} and z_{pik} for each $p \in [3], i \in [n]$, and $k \in [e]$, then applying the Big-M method [48, Chapter 3.10]. For simplification, we do not explicitly show the constraints on the tropical principal components $D^{(1)}, D^{(2)}, D^{(3)}$ to be distinct; this can be proved by applying the Big-M method twice. □

Remark 5.23. *Projecting onto a tropical polytope is relatively straightforward compared to projecting onto a tropical linear space. In theory, one could attempt to reformulate the Stiefel tropical linear space optimization problem from Section 5.2 as in Proposition 5.22; however, the increased complexity of the linear space projection map makes this impractical.*

Due to the large number of variables and constraints involved in Proposition 5.22, we are able to directly solve only relatively small cases like Example 5.27.

Heuristic approximation

As noted above, the number of variables in the mixed integer linear programming problem in Proposition 5.22 increases quickly with the number of leaves and data points. Because solving mixed linear integer programming is NP-hard [61], this problem is difficult to solve in practice. In analogy with Algorithm 5.12, therefore, we develop a heuristic method for approximating the optimal solution for the problem in Proposition 5.22.

Algorithm 5.24 (Approximation for the second order PCA as a tropical polytope).

```
Require: A dataset $D^{(i)}$ of points in the tropical projective torus
Ensure: A tropical polytope $\mathcal{P}$ close to $D^{(i)}$
1: Fix an ordered set $V = (D^{(1)}, D^{(2)}, D^{(3)})$ and compute $\mathcal{P} = tconv(V)$.
2: repeat
3:     Sample three datapoints $D^{(j_1)}, D^{(j_2)}, D^{(j_3)}$ randomly from the set of all datapoints.
4:     Let $V' = \{D^{(j_1)}, D^{(j_2)}, D^{(j_3)}\}$.
5:     Compute $d(\mathcal{P}') = d(tconv(V'))$.
6:     if $d(\mathcal{P}) > d(\mathcal{P}')$, set $V \leftarrow V'$.
7: until convergence.
```

As before, convergence can be assessed by considering whether a new choice of V has been found over a fixed number of previous iterations. If computational time is limited, another approach might simply be to prespecify a total number t of samples. Of course, when the computational cost is reasonable one could enumerate through all $\binom{n}{3}$ different choices for the generating points of $\mathcal{P}$ instead of sampling.

Remark 5.25. *Three data points $D^{(j_1)}, D^{(j_2)}$, and $D^{(j_3)}$ define both a Stiefel tropical linear space L_p and a tropical polytope $\mathcal{P}$. Because Stiefel tropical linear spaces are tropically convex, and each of the generating points is contained in L_p, we see that $\mathcal{P} \subseteq L_p$. In particular, given the same convergence criteria, we should expect Algorithm 5.12 to provide a somewhat better fit than Algorithm 5.24.*

Remark 5.26. *Note that Algorithm 5.24 is well-suited for applications to phylogenetics. Because $\mathcal{U}_m$ is a tropical linear space (Theorem 1.29) and tropical linear spaces are tropically convex, the solution set $\mathcal{P} = tconv(D^{(1)}, D^{(2)}, D^{(3)})$ obtained from Algorithm 5.24 will be contained in the space of ultrametrics. In particular, projections of ultrametrics are also ultrametrics.*

5.5 Simulations

In this section, we apply the previous results to simulated ultrametric datasets.

Exact methods

We begin by identifying the exact best-fit tropical polytope with three vertices closest to a small dataset of equidistant trees using Proposition 5.22. We implemented this proposition using an `R` interface to the popular optimization software `IBM ILOG CPLEX`, called `cplexAPI`.

Example 5.27. *We randomly generated 8 equidistant trees by* `rcoal()` *function in the* `R` *package* `ape`*, with 5 leaves $(t_1, t_2, t_3, t_4, t_5)$ and computed their vectorized distance matrices in Table 5.1.*

tree1	0.2568	0.2314	0.8480	0.2499	0.2568	0.8480	0.2568	0.8480	0.2499	0.8480
tree2	0.0721	1.5059	2.4214	1.5059	1.5059	2.4214	1.5059	2.4214	0.4248	2.4214
tree3	0.4002	0.1322	7.9888	1.4365	0.4002	7.9888	1.4365	7.9888	1.4365	7.9888
tree4	0.0121	2.2597	0.3387	2.2597	2.2597	0.3387	2.2597	2.2597	1.0985	2.2597
tree5	0.4444	2.3015	1.7791	0.7143	2.3015	1.7791	0.7143	2.3015	2.3015	1.7791
tree6	0.3201	2.0053	2.5919	2.5919	2.0053	2.5919	2.5919	2.5919	2.5919	0.6206
tree7	0.1512	3.0181	3.0181	3.0181	3.0181	3.0181	3.0181	0.4858	0.4858	0.0617
tree8	0.2645	1.6765	1.1412	1.6765	1.6765	1.1412	1.6765	1.6765	0.4770	1.6765

Table 5.1: Vectorized Distance Matrices of the Simulated Trees

$D^{(1)}$	0.5894	2.0232	2.0232	2.0232	2.0232	2.0232	2.0232	0.9241	0.9241	0.5000
$D^{(2)}$	0.5000	2.3571	1.5973	0.7698	2.3571	1.5973	0.7698	2.3571	2.3571	1.2237
$D^{(3)}$	0.7679	0.5000	8.3565	1.8042	0.7679	8.3565	1.8042	8.3565	1.8042	8.3565

Table 5.2: Vectorized Distance Matrices of the 2nd PCs

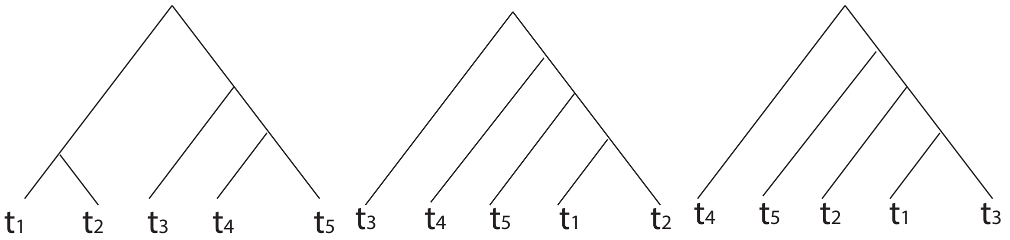


Figure 5.6: $D^{(1)}, D^{(2)}, D^{(3)}$ for Example 5.27

Using our optimization problem formulation from Proposition 5.22, we obtain principal components $D^{(1)}, D^{(2)}, D^{(3)}$ for this example. These points are ultrametrics, and they are described in Figure 5.6 and Table 5.2. In this case the best-fit tropical polytope obtains the sum of distances to be 6.208443. The running time of this exact algorithm was about 50 seconds.

Approximative algorithms

To analyze larger datasets, we turn to the approximative Algorithms 5.12 and 5.24. We implemented both algorithms in `R`.[1] We then generated a random sample from `Mesquite` [64] and applied our algorithms on this dataset. The sample was constructed as follows:

[1] Our software for all computations can be downloaded at `http://polytopes.net/computations/tropicalPCA/`.

Algorithm 5.28 (Generating the simulation dataset). *1. In* `Mesquite`*, generate a taxa block association between 8 species taxa and 8 gene taxa;*

2. Generate a random species tree with the above 8 species taxa via uniform speciation (Yule) simulated tree, where the tree depth (number of generations) is set to be 100000;

3. Generate 250 equidistant gene trees with the 8 gene taxa in first step contained in the above species tree by the coalescence model, where the effective population size is set to be 100000;

4. Compute ultrametric distances from the gene trees by the package `ape` *in* `R`.

After generating trees via Algorithm 5.28, we compute approximate second order tropical principal components via Algorithms 5.12 and 5.24.

The trees in this simulated dataset are well-dispersed with respect to tree topology. Most trees have unique tree topologies and it is therefore difficult to summarize the characteristics of this dataset based on the original trees. We applied both methods of tropical principal component analysis to this set of random equidistant trees generated by Algorithm 5.28. In analogy with [84], we define summary statistics to describe the fit of a Stiefel tropical linear space or a tropical polytope to a given data set. If L_p is a Stiefel tropical linear space, we define its distance to the datapoints $d(L_p)$ as

$$d(L_p) = \sum_i d(D^{(i)}, L_p),$$

and a tropical proportion of variance statistic

$$r(L_p) = \frac{\sum_i d_{tr}(\bar{\pi}, \pi_{L_p}(D^{(i)}))}{\sum_i d_{tr}(D^{(i)}, \pi_{L_p}(D^{(i)})) + \sum_i d_{tr}(\bar{\pi}, \pi_{L_p}(D^{(i)}))}$$

where $\bar{\pi}$ denotes a Fermat-Weber point of the projections of the datapoints, as in [62]. These statistics are defined analogously for a tropical polytope $\mathcal{P}$. The statistic $r(L_p)$ can be interpreted as the proportion of variance explained by L_p; in order to remain consistent with the tropical metric, we sum distances rather than squared distances.

For the polytopal approach, as noted above, the projections will remain ultrametrics. We therefore analyze the topologies of these projections, and compare them with the topology of the species tree.

Approximation results

We applied Algorithm 5.12 to find an approximate 2-dimensional best-fit Stiefel tropical linear space with a convergence threshold of 100 iterations. This execution took about 21 hours to finish, running in parallel on sixteen CPU cores of a server operated by the Mathematics Department of UC Berkeley. The summary statistics for this run were: $d(L_p) = 188.2556$ and $r(L_p) = 0.376$.

We also applied a variant of Algorithm 5.24 to find an approximate best-fit tropical polytope with three vertices, in which we enumerated through all $\binom{250}{3}$ different choices. It was run on macOS with a 2.7 GHz Intel Core i5 dual-core processor, and took about 26 hours to finish. The summary statistics were: $d(\mathcal{P}) = 220.3286$ and $r(\mathcal{P}) = 0.374$. Note that the best-fit Stiefel tropical linear space attained a somewhat closer fit, though the two methods explained similar proportions of variance.

For the tropical polytope method, we recall that projections of equidistant trees will remain ultrametrics. We present common topologies of the projections as well as the species tree topology in Figure 5.7.[2] We observe that these topologies of projected trees are broadly consistent with the topology of the species tree under which these gene trees were generated: taxa g, e and b group together, as do taxa c, h and f. We can view our best-fit tropical polytope as preserving these features of the species tree, meaning that this tropical polytope retains information after projection.

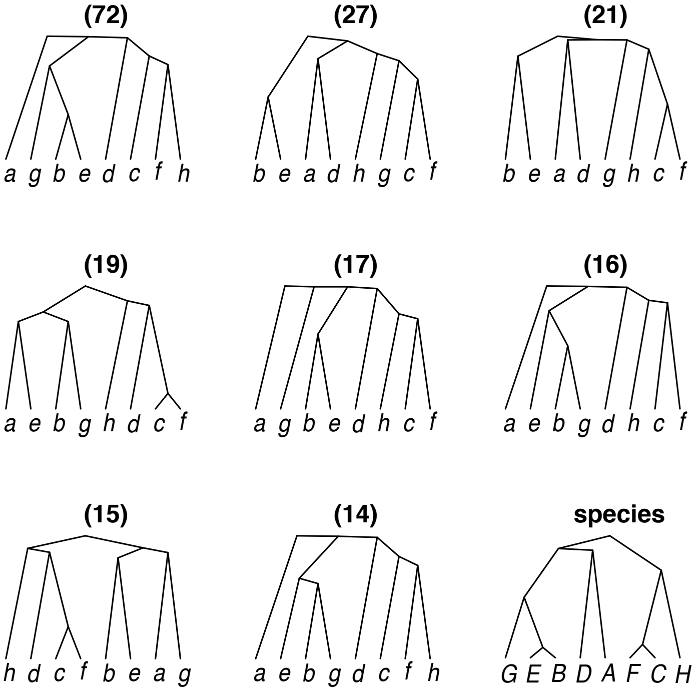


Figure 5.7: Topology frequencies after projections: the parenthesized numbers are frequencies, and the last tree gives the species tree topology.

[2] In keeping with the format of `Mesquite` [64], leaves of the species tree are rendered as capital letters. Tree topologies of all projected points can be found in the supplement at `http://polytopes.net/computations/tropicalPCA/`.

5.6 Apicomplexa genome

We also applied our tropical principal component algorithms to a set of equidistant trees constructed from 252 orthologous sequences on eight species of protozoa in the Apicomplexa phylum by [57]. This dataset was analyzed by Weyenberg et. al; one can find more details, including the gene sequences, in [87]. Because ordinary PCA is sensitive to outliers, we removed 16 outlier gene trees identified by [87] before fitting the tropical principal components.

To find an approximate best-fit 2-dimensional Stiefel tropical linear space, we applied Algorithm 5.12 with a convergence threshold of 100 iterations. Due to the stochastic nature of the algorithm, we executed the algorithm three times. Each execution was run in parallel on sixteen CPU cores of a server operated by the Mathematics Department of UC Berkeley, and took about eighteen hours to finish. The summary statistics remained consistent between these runs. For one representative execution, these statistics were: $d(L_p) = 145.38$ and $r(L_p) = 0.616$.

We also applied a variant of Algorithm 5.24 to find a well-fitted tropical polytope with three vertices, enumerating through all $\binom{252}{3}$ possibilities. It was run on macOS with a 2.7 GHz Intel Core i5 dual-core processor, and took about 16 hours to finish. The summary statistics for this run were: $d(\mathcal{P}) = 147.0568$ and $r(\mathcal{P}) = 0.612$. We note that these summary statistics are relatively consistent with the summary statistics obtained from the Stiefel tropical linear space algorithm.

The tree topologies are presented in Figure 5.8. In general, the projected topologies were largely congruent with the generally accepted phylogeny: the two Plasmodium species (Pv and Pf) group together, as do the four species Ta, Bb, Tg, and Et, and Tt is isolated on a deep branch.

[22, Theorem 23] tells us the tropical convex hull of the rows and columns of a matrix are equal. This allows us to visualize our best-fit tropical polytope in the two-dimensional plane $\mathbb{R}^3/\mathbb{R}\,\mathbf{1}$ as the tropical convex hull of 28 points. These 28 points divide the polytope into different cells, as described in [51, Example 9]. We plot this polytope, along with its cells and the projections of our data points, in Figure 5.9, which functions as a simple two-dimensional visual summary of the clustering of our data, as an ordinary 2-PCA might provide. We note that the different topologies divide the tropical polytope PCA into several regions of positive area, as suggested by Theorem 5.17. Furthermore, the boundaries between these regions often seem to correspond to single tree rearrangement operations: for example, the black- and red-labeled topologies differ by a single nearest-neighbor interchange operation.

In [84], Nye et. al applied the locus of weighted Fréchet mean with the the Billera-Holmes-Vogtman (BHV) [6] metric to apicomplexa data set. The tree topology 1 in Figure 5 of [84] with BHV metric (20 trees) and the tree topology with purple (21 trees) in Fig. 5.8 with the tropical metric have the same tree topology. The tree topology 4 (175 trees) in Figure 5 of [84] with BHV metric and the tree topology with green (70 trees) in Fig. 5.8 with the tropical metric also have the same tree topology, as do the tree topology 2 (9 trees) in Figure 5 of [84] with BHV metric and our tree topology with blue (27 trees) in

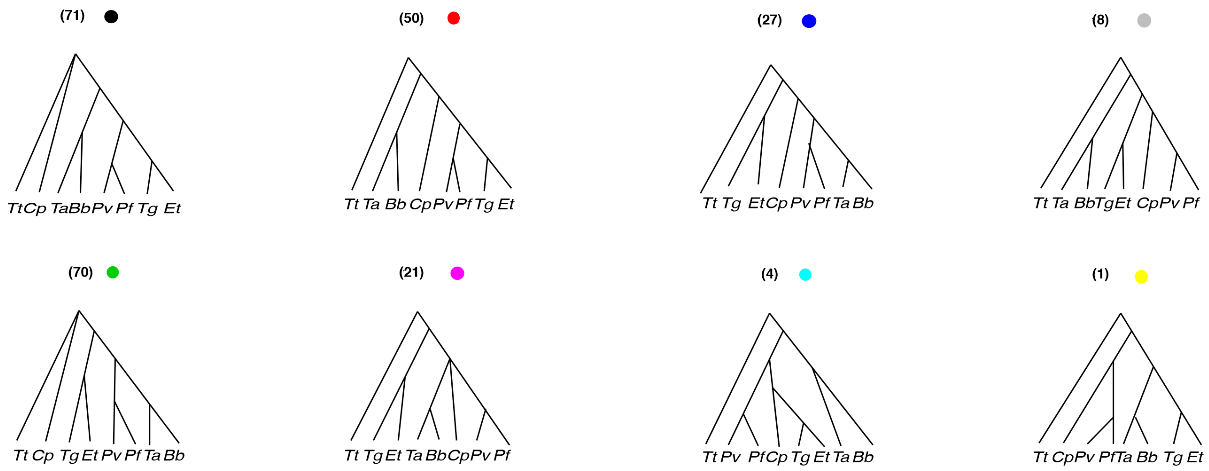


Figure 5.8: Projected topology frequencies from the Apicomplexa dataset: parenthesized numbers give the frequencies of each topology, while the color labels are used in Figure 5.10 below.

Fig. 5.8. In addition, the tree topology 6 (45 trees) in Figure 5 of [84] with BHV metric and the tree topology with black (71 trees) in Fig. 5.8 with the tropical metric have the same tree topology with a different location of the root. Also note that one of the second order principal components (PC3) for the Apicomplexa data set in Figure 5.10 has the same tree topology with the Fréchet mean under the BHV metric for the data set [84], with a different location of the root. As noted above, the tree topologies in each region of our tropical projection often seem to differ by a nearest neighbor interchange move. This also seemed to hold in [84].

The r^2 statistic for the BHV metric is 56% in [84]; our similarly-defined r statistic with the tropical metric is 61.6% with Algorithm 5.12 and 61.2% with Algorithm 5.24. Compared with the results in [84], projected points from each data point onto our tropical polytope seem to be more spread out than they are in the locus of weighted Fréchet mean with the the BHV metric. It is not clear why there are several wide open areas in the locus of weighted Fréchet mean with the BHV metric, whereas this is not in the case with the tropical metric.

Conclusion

In this chapter we discussed tropical analogues to principal component analysis which are well-suited for phylogenetic tree datasets. We proved that the tropical principal hyperplane to e points in $\mathbb{R}^e / \mathbb{R}\mathbf{1}$ attains a total distance from the points equal to their tropical volume. We also showed that the cellular decomposition of any tropical polytope spanned

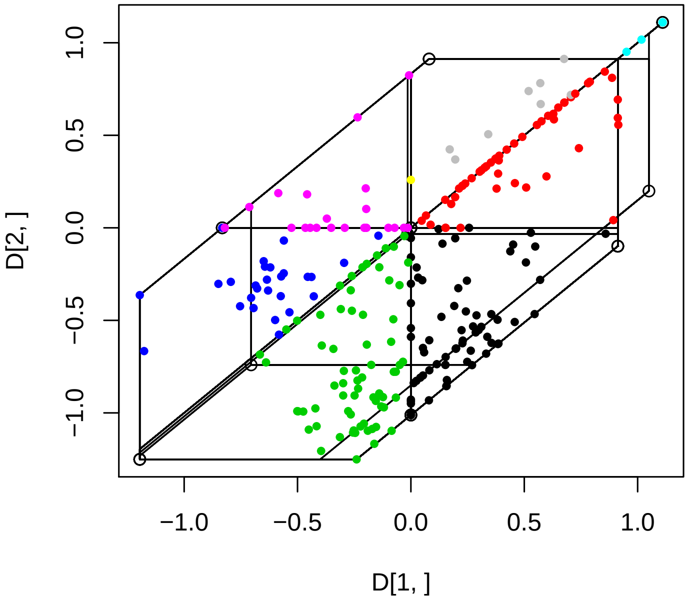


Figure 5.9: Projected points in the tropical polytope PCA, colored as in Figure 5.8.

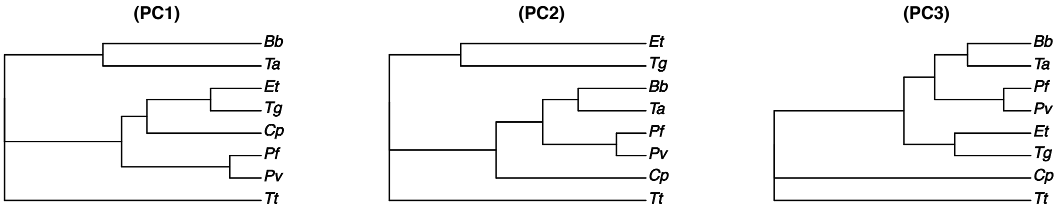


Figure 5.10: The second order PC for the Apicomplexa data set.

by ultrametrics refines the decomposition of the polytope into different tree topologies. We also described heuristic algorithms for calculating these tropical principal components and applied them to study a real-world dataset. Future work to further develop tropical principal component analysis could attempt to demonstrate the existence of tropical principal polytopes containing a tropical Fermat-Weber point of the data points.

Bibliography

[1] P. Abramenko and K. Brown. *Buildings: Theory and Applications.* Graduate Texts in Mathematics. New York: Springer-Verlag, 2008.

[2] X. Allamigeon et al. "Log-barrier interior point methods are not strongly polynomial". In: *SIAM J. Appl. Algebra Geom.* 2.1 (2018), pp. 140–178.

[3] E. Baldwin and P. Klemperer. "Understanding preferences: demand types and the existence of equilibrium with indivisibilities". In: *Econometrica* 87.3 (2019), pp. 867–932.

[4] L. Bernardin et al. *Maple Programming Guide.* Waterloo ON, Canada: Maplesoft, a division of Waterloo Maple Inc., 1996-2020.

[5] D. I. Bernstein. "L-infinity optimization to Bergman fans of matroids with an application to phylogenetics". In: *SIAM J. Discrete Math.* (2020).

[6] L. Billera, S. Holmes, and K. Vogtman. "Geometry of the space of phylogenetic trees". In: *Advances in Applied Mathematics* 27 (2001), pp. 733–767.

[7] T. Bogart et al. "Computing tropical varieties". In: *J. Symbolic Comput.* 42.1-2 (2007), pp. 54–73.

[8] W. Bosma, J. Cannon, and C. Playoust. "The Magma algebra system. I. The user language". In: *J. Symbolic Comput.* 24.3-4 (1997). Computational algebra and number theory (London, 1993), pp. 235–265.

[9] T. Brysiewicz. "Numerical software to compute Newton polytopes and tropical membership". In: *Math. Comput. Sci.* 14 (2020), pp. 577–589.

[10] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic Complexity Theory.* English. Vol. 315. Berlin: Springer, 1997, pp. xxiii + 618.

[11] R. Burkard, M. Dell'Amico, and S. Martello. *Assignment Problems.* Philadelphia, PA: Society for Industrial and Applied Mathematics, 2009.

[12] D. Cartwright et al. "Mustafin varieties". In: *Selecta Math.* 17 (2011), pp. 757–793.

[13] J. Cassels and E. Flynn. *Prolegomena to a Middlebrow Arithmetic of Curves of Genus 2.* Vol. 230. London Mathematical Society Lecture Note Series. Cambridge University Press, 1996.

[14] A. Chan. “Gröbner bases over fields with valuation and tropical curves by coordinate projections”. PhD thesis. University of Warwick, 2013.

[15] M. Chan and B. Sturmfels. “Elliptic curves in honeycomb form, *in* Algebraic and Combinatorial Aspects of Tropical Geometry”. In: *Contemp. Math.* 589 (2013), pp. 87–107.

[16] M. Chan, S. Galatius, and S. Payne. “Tropical curves, graph complexes, and top weight cohomology of $\mathcal{M}_g$”. In: *J. Amer. Math. Soc.* 34 (2021), pp. 565–594.

[17] J. Chen, S. Vemulapalli, and L. Zhang. “Computing unit groups of curves”. In: *J. Symb. Comput.* 104 (2021), pp. 236–255.

[18] H. Cohen. *A Course in Computational Algebraic Number Theory.* Graduate Texts in Mathematics. Heidelberg: Springer-Verlag, 1993.

[19] D. Cox, J. Little, and D. O’Shea. *Ideals, Varieties, and Algorithms.* Fourth. Undergraduate Texts in Mathematics. Switzerland: Springer-Verlag, 2015.

[20] W. Decker et al. SINGULAR *4-1-1 — A computer algebra system for polynomial computations.* `http://www.singular.uni-kl.de`. 2018.

[21] J. Depersin, S. Gaubert, and M. Joswig. “A tropical isoperimetric inequality”. In: *Séminaire Lotharingien de Combinatoire 78B* (2017), p. 12.

[22] M. Develin and B. Sturmfels. “Tropical convexity”. In: *Doc. Math.* 9 (2004), pp. 1–27.

[23] A. Dickenstein and I. Z. Emiris, eds. *Solving Polynomial Equations.* Vol. 14. Algorithms and Computation in Mathematics. Foundations, algorithms, and applications. Springer-Verlag, Berlin, 2005, pp. xiv+425.

[24] A. Dress and W. Terhalle. “The tree of life and other affine buildings”. In: *Proceedings of the International Congress of Mathematicians, Vol. III (Berlin, 1998).* Extra Vol. III. 1998, pp. 565–574.

[25] D. Eisenbud. *Commutative Algebra with a View Towards Algebraic Geometry.* First. Graduate Texts in Mathematics. New York: Springer-Verlag, 1995.

[26] A.-S. Elsenhans and J. Jahnel. “The discriminant of a cubic surface”. In: *Geom. Dedicata* 159 (2012), pp. 29–40.

[27] G. Faltings. “Toroidal resolutions for some matrix singularities”. In: *Moduli of abelian varieties (Texel Island, 1999).* Vol. 195. Progr. Math. Birkhäuser, Basel, 2001, pp. 157–184.

[28] J. C. Faugère et al. “Efficient computation of zero-dimensional Gröbner bases by change of ordering”. In: *J. Symbolic Comput.* 16.4 (1993), pp. 329–344.

[29] A. Feragen et al. “Tree-space statistics and approximations for large-scale analysis of anatomical trees”. In: *IPMI 2013: Information Processing in Medical Imaging* (2012).

[30] A. Fink and F. Rincón. “Stiefel tropical linear spaces”. In: *J. Combin. Theory A* 135 (2015), pp. 291–331.

[31] E. Flynn and N. Smart. “Canonical heights on the Jacobians of curves of genus 2 and the infinite descent”. In: *Acta Arith.* 79 (4 June 1997), pp. 333–352.

[32] B. Fontaine, J. Kamnitzer, and G. Kuperberg. “Buildings, spiders, and geometric Satake”. In: *Compos. Math.* 149.11 (2013), pp. 1871–1912.

[33] D. Ford and O. Veres. “On the complexity of the Montes ideal factorization algorithm”. In: *Algorithmic Number Theory*. Ed. by G. Hanrot, F. Morain, and E. Thomé. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 174–185.

[34] L. Fuchs. *Abelian Groups*. International Series of Monographs on Pure and Applied Mathematics. New York: Pergamon Press, 1960.

[35] J. von zur Gathen and J. Gerhard. *Modern computer algebra*. Third. Cambridge University Press, Cambridge, 2013, pp. xiv+795.

[36] E. Gawrilow and M. Joswig. “POLYMAKE: a framework for analyzing convex polytopes”. In: *in Polytopes: combinatorics and computation* (2000). DMV Seminar 29, Birkhäuser, Basel, pp. 43–73.

[37] P. Görlach, Y. Ren, and L. Zhang. “Computing zero-dimensional tropical varieties via projections”. In: submitted to *Comput. Complex.* (2021).

[38] D. R. Grayson and M. E. Stillman. *Macaulay2, a software system for research in algebraic geometry*. `http://www.math.uiuc.edu/Macaulay2/`. 2018.

[39] G.-M. Greuel and G. Pfister. *A* SINGULAR *introduction to commutative algebra*. With contributions by Olaf Bachmann, Christoph Lossen and Hans Schönemann. Springer-Verlag, Berlin, 2002, pp. xviii+588.

[40] M. A. Hahn and B. Li. “Mustafin varieties, moduli spaces and tropical geometry”. In: *Manuscripta Math.* (2021).

[41] S. Hampe. “Tropical linear spaces and tropical convexity”. In: *Electr. J. Comb.* 22 (2015).

[42] M. Hampton and A. Jensen. “Finiteness of spatial central configurations in the five-body problem”. In: *Celestial Mech. Dynam. Astronom.* 109.4 (2011), pp. 321–332.

[43] M. Hampton and R. Moeckel. “Finiteness of relative equilibria of the four-body problem”. In: *Invent. Math.* 163.2 (2006), pp. 289–312.

[44] K. Hept and T. Theobald. “Tropical bases by regular projections”. In: *Proc. Amer. Math. Soc.* 137.7 (2009), pp. 2233–2241.

[45] P. Hitzelberger. “Non-discrete affine buildings and convexity”. In: *Adv. Math.* 227.1 (2011), pp. 210–244.

[46] T. Hofmann and Y. Ren. “Computing tropical points and tropical links”. In: *Discrete Comput. Geom.* 60.3 (2018), pp. 627–645.

[47] D. Holmes. “Computing Néron–Tate heights of points on hyperelliptic Jacobians”. In: *J. Number Theory* 132 (6 June 2012), pp. 1295–1305.

[48] G. Igor, N.G. Stephan, and S. Ariela. *Linear and Nonlinear Optimization.* 2nd ed. Society for Industrial Mathematics, 2009.

[49] A. Jensen, Y. Ren, and F. Seelisch. `gfan.lib`. *A* SINGULAR *4-1-2 interface to gfanlib and more.* `http://www.singular.uni-kl.de`. 2017.

[50] A. N. Jensen. *Gfan 0.6.2, a software system for Gröbner fans and tropical varieties.* Available at `http://home.imf.au.dk/jensen/software/gfan/gfan.html`. 2017.

[51] M. Joswig, B. Sturmfels, and J. Yu. "Affine buildings and tropical convexity". In: *Albanian J. Math.* 1 (2007), pp. 187–211.

[52] M. Joswig. *Essentials of Tropical Combinatorics*. Graduate Studies in Mathematics. `http://page.math.tu-berlin.de/~joswig/etc/ETC-210408.pdf`. American Mathematical Society, to be published.

[53] M. Joswig and K. Kulas. "Tropical and ordinary convexity combined". In: *Adv. Geom.* 10 (2010), pp. 333–352.

[54] M. Joswig and B. Schröter. "The degree of a tropical basis". In: *Proc. Amer. Math. Soc.* 146.3 (2018), pp. 961–970.

[55] M. Kapovich, B. Leeb, and J. J. Millson. "The generalized triangle inequalities in symmetric spaces and buildings with applications to algebra". In: *Mem. Amer. Math. Soc.* 192.896 (2008), pp. viii+83.

[56] S. Keel and J. Tevelev. "Geometry of Chow quotients of Grassmannians". In: *Duke Math. J.* 134.2 (2006), pp. 259–311.

[57] C. Kuo, J. P. Wares, and J. C. Kissinger. "The Apicomplexan whole-genome phylogeny: An analysis of incongruence among gene trees". In: *Mol. Biol. Evol.* 25 (2008), pp. 2689–2698.

[58] Y. N. Lakshman. "A single exponential bound on the complexity of computing Gröbner bases of zero-dimensional ideals". In: *Effective methods in algebraic geometry (Castiglioncello, 1990).* Vol. 94. Progr. Math. Birkhäuser Boston, Boston, MA, 1991, pp. 227–234.

[59] Y. N. Lakshman and D. Lazard. "On the complexity of zero-dimensional algebraic systems". In: *Effective methods in algebraic geometry (Castiglioncello, 1990).* Vol. 94. Progr. Math. Birkhäuser Boston, Boston, MA, 1991, pp. 217–225.

[60] D. Lazard. "Solving zero-dimensional algebraic systems". In: *J. Symbolic Comput.* 13.2 (1992), pp. 117–131.

[61] H. W. Lenstra. "Integer Programming with a fixed number of Variables". In: *Mathematics of Operations Research* 8 (1983), pp. 538–548.

[62] B. Lin and R. Yoshida. "Tropical Fermat-Weber points". In: *SIAM J. Discrete Math.* 32 (2018), pp. 1229–1245.

[63] D. Maclagan and B. Sturmfels. *Introduction to Tropical Geometry.* Vol. 161. Graduate Studies in Mathematics. Providence, RI: American Mathematical Society, 2015.

[64] W. P. Maddison and D. R. Maddison. *Mesquite: a modular system for evolutionary analysis.* Version 3.31 http://mesquiteproject.org. 2017.

[65] K. Mahler. “On Minkowski’s theory of reduction of positive definite quadratic forms”. In: *Q. J. Math.* 9 (1 Jan. 1938), pp. 259–262.

[66] T. Markwig and Y. Ren. “Computing tropical varieties over fields with valuation”. In: *Found. Comput. Math.* (Aug. 2019).

[67] V. Miller. “Short programs for functions on curves”. In: (May 1986). , `https://crypto.stanford.edu/miller/miller.pdf`, unpublished.

[68] A. Monod et al. “Tropical foundations for probability & statistics on phylogenetic tree space”. In: (ePrint: `arXiv:1805.12400`, 2018).

[69] J. Müller. “Computing canonical heights using arithmetic intersection theory”. In: *Math. Comp.* 83.285 (May 2013), pp. 311–336.

[70] J. Müller and M. Stoll. “Computing canonical heights on elliptic curves in quasi-linear time”. In: *LMS J. Comput. Math.* 19 (A Aug. 2016), pp. 391–405.

[71] J. Neukirch. *Algebraic Number Theory.* Vol. 322. Grundlehren der Mathematischen Wissenschaften. Springer-Verlag, Berlin, 1999.

[72] M. Panizzut and M. D. Vigeland. *Tropical lines on cubic surfaces.* `arXiv:0708.3847v2`. 2019.

[73] G. Pappas, M. Rapoport, and B. Smithling. “Local models of Shimura varieties, I. Geometry and combinatorics”. In: *Handbook of moduli. Vol. III.* Vol. 26. Adv. Lect. Math. (ALM). Int. Press, Somerville, MA, 2013, pp. 135–217.

[74] J. Richter-Gebert, B. Sturmfels, and T. Theobald. “First steps in tropical geometry”. In: Idempotent Mathematics and Mathematical Physics 377 (2005). Ed. by G. Litvinov and e. V. Maslov, pp. 289–308.

[75] F. Rincón. “Local tropical linear spaces”. In: *Discrete Comput. Geom.* 50.3 (2013), pp. 700–713.

[76] M. Rosenlicht. “Some rationality questions on algebraic groups”. In: *Ann. Mat. Pura Appl.* 43 (1957), pp. 25–50.

[77] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 8.2).* `http://www.sagemath.org`. 2018.

[78] P. Samuel. “A propos du théoreme des unités”. In: *Bull. des Sci. Math.* 90 (1966), pp. 89–96.

[79] J. H. Silverman. *The Arithmetic of Elliptic Curves.* Second. Graduate Texts in Mathematics. New York: Springer-Verlag, 2009.

[80] A. Steenpass. `modular.lib`. *A* Singular *4-1-2 library for modular techniques.* 2019.

[81] M. Stoll. "An explicit theory of heights for hyperelliptic Jacobians of genus three". In: *Algorithmic and Experimental Methods in Algebra, Geometry, and Number Theory.* Ed. by G. Böckle, W. Decker, and G. Malle. Springer-Verlag, 2017, pp. 665–715.

[82] B. Sturmfels. *Solving systems of polynomial equations.* Vol. 97. CBMS Regional Conference Series in Mathematics. American Mathematical Society, Providence, RI, 2002.

[83] T. Supasiti. "Serre's tree for $SL_2(\mathbb{F})$". Honors thesis. University of Melbourne, 2008.

[84] T. Nye et al. "Principal component analysis and the locus of the Fréchet mean in the space of phylogenetic trees". In: *Biometrika* (2017), pp. 901–922.

[85] J. Tevelev. "Compactifications of subvarieties of tori". In: *Am. J. Math.* 129.4 (2007), pp. 1087–1104.

[86] N. M. Tran and J. Yu. "Product-mix auctions and tropical geometry". In: *Math. Oper. Res.* 44.4 (2019), pp. 1396–1411.

[87] G. Weyenberg, R. Yoshida, and D. Howe. "Normalizing kernels in the Billera-Holmes-Vogtmann treespace". In: *IEEE/ACM Trans. Comput. Biol. Bioinform.* (2016).

[88] H. Weyl. "Theory of reduction for arithmetical equivalence". In: *Trans. Amer. Math. Soc.* 48 (1940), pp. 126–164.

[89] R. Yoshida, R. Page, and L. Zhang. "Tropical principal component analysis on the space of phylogenetic trees". In: *Bioinform.* 36 (2021), pp. 4590–4598.

[90] R. Yoshida, L. Zhang, and X. Zhang. "Tropical principal component analysis and its application to phylogenetics". In: *Bull. Math. Biol.* 81 (2019), pp. 568–597.

[91] L. Zhang. "Computing min-convex hulls in the affine building of SL_d". In: *Discrete Comput. Geom.* 65 (2021), pp. 1314–1336.