Toric and Semitoric Packing Capacities
Final Report

Faculty Mentor: Joseph Palmer
Graduate Student Mentor: Jason Liu
Student: Yu Du, Gabriel Kosmacher,
Jeff Massman, Zheyu Zhang

December 2021

1 Introduction

Delzant and semitoric polygons are the 2-dimensional image under a mapping of a related symplectic toric and symplectic semitoric 4-manifolds, respectively. The existence and uniqueness of these polygons allow for symplectic capacities, a topic of much interest in high dimensional symplectic geometry, to be computed in lower dimensions. One instance of this is the packing capacity problem. This paper discusses interesting results surrounding 2-dimensional packing and presents an algorithm to solve for the packing capacity problem for any Delzant polygon with exact arithmetic. This type of packing for Delzant polygons first appeared in [2], and the semitoric case first appeared in [1].

2 Background

In this section, we will introduce the background information necessary for understanding the problem of packing Delzant and semitoric polygons. This will include relevant definitions, examples, and a formal exposition of the problem statement.

2.1 Delzant and Semitoric Polygons

Put simply, Delzant and semitoric polygons are 2-dimensional polygons that satisfy a set of special conditions. To fully understand exactly what Delzant and semitoric polygons are, a comprehensive understanding of these conditions is imperative. Therefore, we will explicitly and carefully define all of these conditions before we define what Delzant and semitoric polygons are.

Definition 2.1 (Primitive Vectors in $\mathbb{Z}^2$)**.** *A vector $v \in \mathbb{Z}^2$ is said to be **primitive** if the greatest common divisor of its components is* 1*, i.e. they are coprime.*

In the context of Delzant and semitoric polygons, we are particularly interested in the primitive edge direction vectors, which are just the primitive vectors along the edges of the polygon.

Definition 2.2 (Rational Polygon)**.** *A polygon is said to be **rational** if the slopes of its edges are rational or undefined (i.e. a vertical edge).*

Given any vertex of a rational polygon, the adjacent two edges are directed along unique primitive vectors in $\mathbb{Z}^2$. Given two vectors $v, w \in \mathbb{Z}^2$ let $[v, w]$ be the 2×2 matrix which has v as its first column and w as its second column. We call $\det([v, w])$ the *determinant of* v, w.

Definition 2.3 (Delzant Condition)**.** *A vertex of a polygon is said to satisfy the **Delzant condition** if and only if the determinant of the primitive direction vectors along its adjacent edges is* ± 1*.*

See Figure 1 for an example.

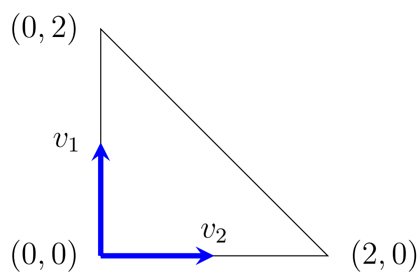


Figure 1: Delzant Condition. The vertex at $(0,0)$ satisfies the Delzant condition because the two primitive edge direction vectors $u_1 = \left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right)$ and $u_2 = \left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right)$ satisfy the condition $\det(u_1, u_2) = \det\left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right) = 1$. The reader can verify that the other two vertices also satisfy the Delzant condition.

With all the background exposited, we may now define Delzant polygons.

Definition 2.4 (Delzant Polygon)**.** *A **Delzant polygon** is a convex rational polygon such that every vertex satisfies the Delzant condition.*

The polygon in Figure 1 is a Delzant polygon: It is convex, rational, and every vertex satisfies the Delzant condition.

Before introducing semitoric polygons, we first define an important matrix essential to their study:

$$\mathrm{T} := \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \tag{1}$$

Remark 2.5. Roughly speaking, this matrix has the effect of incrementing the slope of a vector by one, and unsurprisingly, T^{-1} decrements the slope by one. It is left as an exercise to verify this.

We will see next how this matrix plays an important role in the definition of semitoric polygons.

Definition 2.6 (Semitoric Polygon)**.** *A* ***semitoric polygon*** *is a triple* $(\Delta, \vec{c}, \vec{\epsilon})$ *such that*

1. $\Delta \subset \mathbb{R}^2$ *is a convex rational polygon,*

2. $\vec{c} = (c_1, \ldots, c_m)$ *is the set of* ***marked points****, and* $c_i \in \mathrm{int}(\Delta)$ *for* $i = 1, \ldots, m$ *(i.e. the marked points are in the interior of* Δ*), and*

3. $\vec{\epsilon} = (\epsilon_1, \ldots, \epsilon_m) \in \{\pm 1\}^m$ *is the set of* ***cut directions****.*

Let $\mathcal{L}_i$ *be the ray starting at* c_i *and going up if* $\epsilon_i = +1$ *and down if* $\epsilon_i = -1$*. We call* $\mathcal{L}_i$ *the* i^{th} ***cut****. Furthermore, each vertex of* Δ *(with vectors* $v, w \in \mathbb{Z}^2$ *directing the adjacent edges such that* $\det(v, w) > 0$*) must satisfy one of the following:*

(I) The Delzant condition,

(II) The vertex is on a cut and $\det(\mathrm{T}v, w) = 0$*, or*

(III) The vertex is on a cut, and $\det(\mathrm{T}v, w) = \pm 1$*.*

Remark 2.7. Point (II) is called the “fake corner condition,” and point (III) is called the “hidden corner condition.” Unsurprisingly, vertices that satisfy the former are called ”fake corners” while vertices that satisfy the latter are called ”hidden corners.”

Cuts and marked points are always associated with a vertex, so it often makes sense to refer to such vertices as “marked corners.” We also denote by h the vertical distance of the marked point to the bottom of the polygon. See Figure 2 for an example.

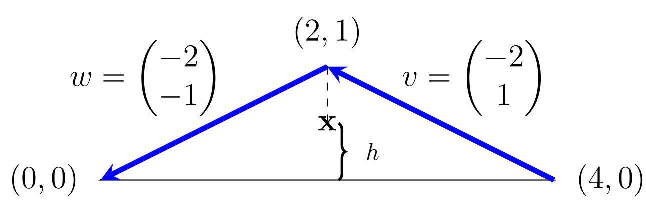


Figure 2: A simple semitoric polygon. The marked point is denoted by the $\mathbf{x}$, the cut is the dashed line, h is the distance from the marked point to the opposite side. The reader should verify that the vertex at $(2,1)$ satisfies the fake corner condition.

The direction of a cut can also be changed: If the cut is up, we can apply the matrix T to the polygon on all the parts right of the cut, whereas if the cut is down, we do the same thing except using T^{-1}. It is also worth noting that any integer power of T can be applied globally to the polygon, and the resulting polygon is considered equivalent to the original.

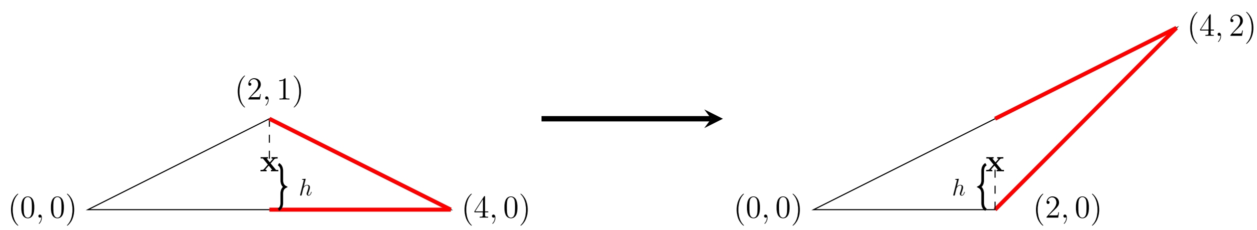


Figure 3: Changing an up cut to a down cut by applying T to the right side (in red) of the polygon. Notice how the vertex originally at $(2, 1)$ disappears after the transformation. This is precisely what is meant by "fake corners."

2.2 Length Measure

When studying Delzant and semitoric polygons, we often care little about the real lengths of edges; rather, we define a new metric to measure length, one that, though initially unintuitive, actually simplifies the packing problem greatly.

Definition 2.8 ($\mathrm{SL}_2(\mathbb{Z})$)**.** *The special linear group of 2 by 2 matrices of integer entries, denoted as* $\mathbf{SL_2(\mathbb{Z})}$*, is a group under the operation of standard matrix multiplication. The defining feature of this group is that all matrices within it have determinant equal to 1.*

Remark 2.9. $\mathrm{SL}_2(\mathbb{Z})$ is not a group of rotations. In fact, the only rotations in $\mathrm{SL}_2\mathbb{Z}$ are net rotations by $\frac{\pi}{2}, \pi, \frac{3\pi}{2}$ and 2π (i.e. 0).

The determinant condition turns out to be a very useful one, as we will see later. The gist is that fixing the determinant to be 1 means that area (and orientation) is preserved under transformation by elements in $\mathrm{SL}_2(\mathbb{Z})$.

Definition 2.10 ($\mathrm{SL}_2(\mathbb{Z})$ Length)**.** *Given a line segment ℓ with rational slope or vertical, the* $\mathbf{SL_2(\mathbb{Z})}$ ***length*** *of ℓ is defined as the real length of the segment after transformation to a coordinate axis by some element of* $\mathrm{SL}_2(\mathbb{Z})$*.*

Remark 2.11. $\mathrm{SL}_2(\mathbb{Z})$ length is well-defined: it is independent of the choice of coordinate axis and matrix used to send the vector to said axis. We will comment on this after we prove the upcoming theorem.

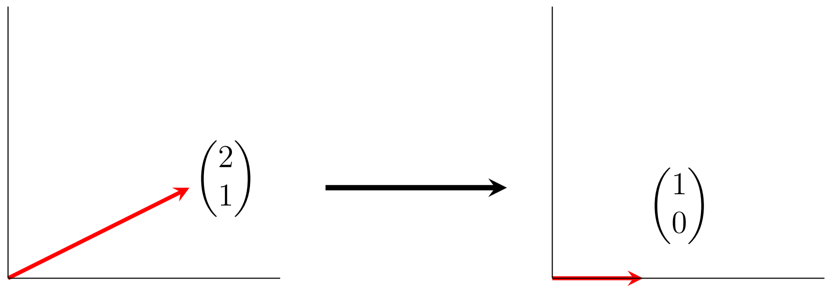


Figure 4: The vector $\binom{2}{1}$ gets sent to $\binom{1}{0}$ by an element in $\mathrm{SL}_2(\mathbb{Z})$. Thus, its $\mathrm{SL}_2(\mathbb{Z})$ length is 1. One such matrix that accomplishes this is $\left(\begin{smallmatrix} 0 & 1 \\ -1 & 2 \end{smallmatrix}\right)$.

This concept is of great importance and is not that intuitive, so we prove an example in Figure 6.

The following theorem gives us a simple way to calculate the $\mathrm{SL}_2(\mathbb{Z})$ length of any permissible vector. First we prove a lemma to aid in the proof of the theorem.

Lemma 2.12. *For $x, y \in \mathbb{R}$, if $\frac{y}{x} \in \mathbb{Q}$, then x and y are either both rational or both irrational such that they can be written, not necessarily uniquely, as rational multiples of the same irrational number.*

Proof. Clearly, if $\frac{y}{x} \in \mathbb{Q}$ it is possible that $x, y \in \mathbb{Q}$, in which case we are done. Now, suppose we only know that x is irrational. Then $x = 1 \cdot x$ and $y = \frac{y}{x} \cdot x$. Similarly, suppose we only know that y is irrational. Then $y = 1 \cdot y$ and $x = \frac{x}{y} \cdot y$ ($y \neq 0$ since it is irrational, and $\frac{x}{y}$ is rational since it is just the reciprocal of $\frac{y}{x}$). In both instances, x and y are rational multiples of the same irrational. This completes the proof. ■

Theorem 2.13. *Take $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^2$ such that $\frac{x_2}{x_1} \in \mathbb{Q}$. Letting $x_1 = c\frac{m}{n}$ and $x_2 = c\frac{p}{q}$ for $c \in \mathbb{R}$ and $p, q, m, n \in \mathbb{Z}$, the $\mathrm{SL}_2(\mathbb{Z})$ length of $\mathbf{x}$ is $\frac{c \, \gcd(p \cdot n, m \cdot q)}{n \cdot q}$.*

Proof. Let $\mathbf{x}$ be a real vector of rational slope. Then by lemma 2.12 we have that $\mathbf{x} = \begin{pmatrix} c\frac{m}{n} \\ c\frac{p}{q} \end{pmatrix}$ for $m, n, p, q \in \mathbb{Z}$ and $c \in \mathbb{R}$. We use a matrix in $\mathrm{SL}_2(\mathbb{Z})$ to send our vector to one in the space $\mathbb{R} \times \{0\}$. Define $d := \gcd(mq, pn)$. Then consider the vector $\begin{pmatrix} \frac{cd}{nq} \\ 0 \end{pmatrix} \in \mathbb{R} \times \{0\}$. Since $\begin{pmatrix} \mathrm{SL}_2(\mathbb{Z}), & \begin{matrix} \text{matrix} \\ \text{multiplication} \end{matrix} \end{pmatrix}$ is a group, it suffices to find $A \in \mathrm{SL}_2(\mathbb{Z})$ such that $A \begin{pmatrix} \frac{cd}{nq} \\ 0 \end{pmatrix} = \mathbf{x}$ as we know $A^{-1} \in \mathrm{SL}_2(\mathbf{Z})$ exists with $A^{-1}\mathbf{x} = \begin{pmatrix} \frac{cd}{nq} \\ 0 \end{pmatrix}$. Let $r = \frac{mq}{d}$ and $k = \frac{pn}{d}$. Clearly, $r, k \in \mathbb{Z}$

and since $d = \gcd(mq, pn)$, we have $\gcd(r, k) = 1$. Thus, $\exists\ s, t \in \mathbb{Z}$ such that $rs + tk = 1$. Consider $A = \begin{pmatrix} r & -t \\ k & s \end{pmatrix} \in \mathrm{SL}_2(\mathbb{Z})$. Then, recalling that $rd = mq$ and $kd = pn$

$$\begin{aligned} A \begin{pmatrix} \frac{cd}{nq} \\ 0 \end{pmatrix} &= \begin{pmatrix} r & -t \\ k & s \end{pmatrix} \begin{pmatrix} \frac{cd}{nq} \\ 0 \end{pmatrix} = \begin{pmatrix} c \cdot \frac{rd}{nq} \\ c \cdot \frac{kd}{nq} \end{pmatrix} \\ &= \begin{pmatrix} c \cdot \frac{(mq)}{nq} \\ c \cdot \frac{(pn)}{nq} \end{pmatrix} = \begin{pmatrix} c\frac{m}{n} \\ c\frac{p}{q} \end{pmatrix} = \mathbf{x} \end{aligned}$$

Thus, A^{-1} sends $\mathbf{x}$ to $\begin{pmatrix} \frac{cd}{nq} \\ 0 \end{pmatrix}$. Therefore, the $\mathrm{SL}_2(\mathbb{Z})$ length of $\mathbf{x}$ is $\frac{cd}{nq} = \frac{c \cdot \gcd(mq, pn)}{nq}$. The argument for when the $\mathrm{SL}_2(\mathbb{Z})$ length of our vector is in the space $\{0\} \times \mathbb{R}$ (i.e. the y-axis) is nearly identical. ■

Corollary 2.14.

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{Z}^2 \implies \mathrm{SL}_2(\mathbb{Z}) \textit{ length of } \mathbf{x} \textit{ is } \gcd(x_1, x_2)$$

Remark 2.15 (Well-defined). Notice how the formula for the $\mathrm{SL}_2(\mathbb{Z})$ length has no dependence on the choice of matrix used; it relies entirely on the components of the vector in question. Furthermore, the formula is the same for both axes. We did not prove the y-axis case explicitly, however the proof is nearly identical; just consider the vector $\begin{pmatrix} 0 \\ \frac{cd}{nq} \end{pmatrix}$.

2.3 Packing Delzant and Semitoric Polygons

An interesting problem in the context of Delzant and semitoric polygons, as well as the central focus of this report, is concerned with "packing" these polygons with triangles in such a way as to satisfy a certain set of conditions. We will define the elements of this problem here.

Definition 2.16 (Packed Triangle)**.** *A **packed triangle** is a triangle placed at a vertex of a Delzant or semitoric polygon such that the triangle is the image under $\mathrm{SL}_2(\mathbb{Z})$ and translation of a right isosceles triangle, called **a model triangle**, given by $\{(x, y) : x \geq 0, y \geq 0, x + y < \lambda\}$, where the vertex of the triangle at $(0, 0)$ gets mapped to the vertex on the polygon, and the legs of the triangle get mapped to the adjacent edges of the vertex of the polygon.*

Definition 2.17 (Packing, Delzant)**.** *A **packing** of a Delzant polygon is a configuration of packed triangles in the polygon such that the triangles are non-intersecting.*

Packing for semitoric polygons is similar, but there are notable differences.

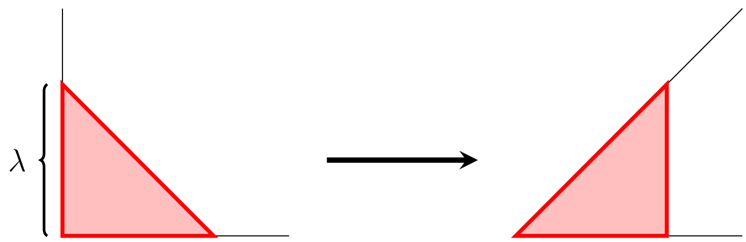


Figure 5: Packed triangle. The left side is the isosceles triangle, while the right side is the image after being sent to a Delzant (or hidden) corner of the polygon in question.

Definition 2.18 (Packing, semitoric)**.** *A **packing** of a* <u>*semitoric*</u> *polygon is a configuration of packed triangles in the polygon such that*

1. *The triangles are non-intersecting*
2. *No triangle is packed at a fake corner, and no triangle contains any marked points*
3. *A single packed triangle is the image of any triangle that is packed for at least one set of cut directions*

Remark 2.19. Point 3 in the semitoric packing definition implies that a semitoric packing can include shapes that are not triangles, but become triangles after changing the cut direction. See figure 6 below for an example of this.

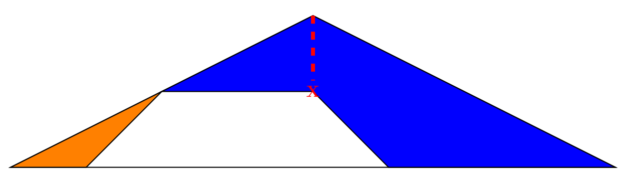


Figure 6: Packing of the semitoric polygon in figure 2. Note that the blue "triangle" is deformed since it is packed at a marked corner; this is because we cannot pack the triangle as-is; we must first transform it using T so that all of its vertices satisfy the Delzant condition. Then we pack the transformed triangle, and when we revert back, the triangle gets deformed. The cut and marked point are colored red for visibility.

As one can imagine, there are a number of possible packings of any given Delzant or semitoric polygon. Indeed, there is an infinite number of them. But there is a certain kind of packing that we care about, which we will introduce next.

Definition 2.20 (Maximum Packing)**.** *A **maximum packing** of a Delzant or semitoric polygon is a packing such that the largest possible aggregate sum of the packed triangle areas that the polygon can sustain is achieved.*

Remark 2.21. Maximum packings are not unique; a Delzant or semitoric polygon can have several maximum packings.

The packing of the semitoric polygon in figure 4 is actually a maximum packing. For examples of maximum packings of Delzant polygons, see section 5 of this report.

Definition 2.22 (Packing Capacity)**.** *The* ***packing capacity*** *of a Delzant or semitoric polygon is the ratio of the total sum of the triangle areas in a maximum packing to the area of the entire polygon.*

Intuitively, the packing capacity is the proportion of the polygon covered by triangles in a maximum packing; in other words, it is a measure of how much the polygon is able to "hold."

Remark 2.23. The capacity only takes values in the interval $[0, 1]$. This gives rise to the notion of a "perfect packing," where the capacity is 1. In reality, very few Delzant or semitoric polygons can achieve this, but some can, for example squares and right isosceles triangles.

3 Theory

From now on, we will only focus on the Delzant case. Let P be a Delzant polygon. We can choose some vertex on P and call it v_1. Then starting from v_1 and going around the polygon counterclockwise on the boundary, we label the vertices by $v_2, ..., v_n, v_{n+1} = v_1$ and label the edge connecting v_i to v_{i+1} by e_i.

The 2-dimensional packing problem is expressed as maximizing the packing area subject to the size of the polygon. To translate this into a mathematical optimization problem, we first note that the area of a triangle is preserved under any action of $\mathrm{SL}_2(\mathbb{Z})$ as the determinant of any matrix in $\mathrm{SL}_2(\mathbb{Z})$ is one. Therefore, the area of ith packed triangle (i.e. the triangle packed at v_i) is $\frac{1}{2}\lambda_i^2$, where λ_i is the leg length of the corresponding model triangle.

Definition 3.1. *Let* $\mathbf{\Lambda} := (\lambda_1, \lambda_2, \ldots, \lambda_n) \in \mathbb{R}^n$ *be the vector containing the* $\mathrm{SL}_2(\mathbb{Z})$ *size of the* n *packed triangles.*

We have an objective function

$$f(\mathbf{\Lambda}) = \frac{1}{2}(\lambda_1^2 + \lambda_2^2 + \cdots + \lambda_n^2) = \frac{1}{2}||\mathbf{\Lambda}||_2^2$$

We note that $\mathbf{\Lambda}$ *represents a valid packing* of the triangles when $\lambda_1, \ldots, \lambda_n$ satisfy the packing conditions on polygon P. A maximum packing is equivalent to maximizing f over the set of all possible *valid* n-tuples $\mathbf{\Lambda}$. We first study this set. Notice that there is a one-to-one correspondence between valid packings and *valid* n-tuples $\mathbf{\Lambda}$, so we will not distinguish them from now on.

Definition 3.2. *Let Δ_λ be the set of all valid packings represented by n-tuples. Set theoretically, we write $\Delta_\lambda = \{\mathbf{\Lambda} \in \mathbb{R}^n : \mathbf{\Lambda}$ represents a valid packing$\}$*

Thus, we want to find the max of f over Δ_λ. We first describe Δ_λ in terms of inequalities. To get a valid packing, we only need to make sure any two packed triangles will not intersect. This boils down to insisting that $\lambda_i + \lambda_{i+1} \leq \ell_i$, where ℓ_i is the $\mathrm{SL}_2(\mathbb{Z})$ length of e_i. Because our packing triangles are regular in $\mathrm{SL}_2(\mathbb{Z})$, this condition ensures that two adjacent packing triangles have size that sums to at most the $\mathrm{SL}_2(\mathbb{Z})$ length of the edge in P they share, enforcing that two packing triangles can never overlap on any edge. With the obvious restriction that length cannot be negative, we have

$$\Delta_\lambda = \{((\lambda_1, \lambda_2, \ldots, \lambda_n)) : \lambda_i \geq 0, \lambda_i + \lambda_{i+1} \leq \ell_i, i = 1, 2, ..., n\}$$

with the same convention that λ_{n+1} means λ_1.

Definition 3.3. *A vertex of Δ_λ is defined to be a 0-dimensional polytope residing at the intersection of at least n affine hyperplanes.*

Lemma 3.4. *The set $\Delta_\lambda \subset \mathbb{R}^n$ is an n-dimensional polytope with at most $\binom{2n}{n}$ vertices.*

Proof. We have $2n$ linear constraints and a vertex lies at the intersection of at least n of these constraints. ■

We now have enough information to express the 2-dimensional packing problem as the mathematical optimization problem

$$\max_{\Delta_\lambda} \frac{1}{2} ||\mathbf{\Lambda}||_2^2. \tag{2}$$

3.1 Main Theorem

Lemma 3.5. *$f(\mathbf{\Lambda})$ restricted to Δ_λ obtains maxima only on the vertices of Δ_λ.*

Proof. We begin by considering the Hessian matrix H of f: $\mathrm{H} = 2\mathrm{I}_n$ which is clearly positive-definite. By the second derivative test, any critical points on the interior of Δ_λ are local minima, and thus the Extreme Value Theorem tells us that maxima of f must reside on the boundary of Δ_λ. Now consider any face of Δ_λ with dimension $0 < m < n$. We call this region $\mathcal{G}$. Consider $f|_\mathcal{G} := f$ restricted to $\mathcal{G}$.

Then $\mathcal{G}$ is a subset of an affine subspace $\mathcal{A}$ (i.e. $\mathcal{G}$ is a linear subspace shifted from the origin). Let v_1 be a unit normal vector of $\mathcal{A}$ and $\mathcal{A} = \mathrm{span}(v_2, ..., v_n) + kv_1$ for some $k \in \mathbb{R}$. Using Gram-Schmidt, we get $v_1, w_2, ..., w_n$ with $\langle v_i, w_j \rangle = 0$. Now for any $\mathbf{c} \in \mathcal{A}$, $c = kv_1 + c_2 w_2 + ... + c_n w_n$. Thus, we can decompose $\mathbf{c}$ into $\mathbf{v} + \mathbf{x}$ where $\mathbf{x} \in \mathcal{G}$ is a vector generating from the head of $\mathbf{v}$ and we have $f|_\mathcal{G}(\mathbf{c}) = ||\mathbf{v}||^2 + \sum_{i=1}^{m} x_i^2$.

Thus we argue similarly to our original case: The Hessian of $f|_{\mathcal{G}}$ with respect to the new coordinates of $\mathbf{x}$ is $2\mathrm{I}_m$, which is positive-definite, and so all critical points on the interior must be minima. Therefore, by the Extreme Value Theorem, maxima of $f|_{\mathcal{G}}$ must occur on the boundary of $\mathcal{G}$. Since this is true for all $0 < m < n$, maxima can only occur on the 0-dimensional facets of Δ_λ, i.e. the vertices.

■

Theorem 3.6. *The 2-dimensional packing problem can be solved exactly in $\mathcal{O}\binom{2n}{n}$ steps.*

Proof. By lemma 3.5, we have that the maximum packing must lie at a corner of Δ_λ. By construction, Δ_λ has at most $\binom{2n}{n}$ vertices. Thus, the solution to the 2-dimensional packing problem is found at the vertex of Δ_λ with maximal magnitude. ■

3.2 Alternating Edge Condition

Remark 3.7. In this scope, we will assign $\lambda_0 := \lambda_n$ and $\ell_0 := \ell_n$. Additionally, $\mathbf{\Lambda}$ will be a packing vector.

Definition 3.8. *We call the constraint $\ell_i \leq \ell_{i+1}$ and $\ell_i \leq \ell_{i-1}$ the **forward and backward alternating edge conditions**, respectively. An alternating edge condition is said to be **fully satisfied** on a subset of edges if each edge in the subset satisfies the forward condition or each edge satisfy the backward condition. Else, the subset does not satisfy the alternating edge condition.*

Lemma 3.9. *Given a $n = 2m$ dimensional Δ_λ,*

$$\max_{\Delta_\lambda} \frac{1}{2}||\mathbf{\Lambda}||_2^2 = \frac{1}{2}\sum_{i\ odd}^{n} \ell_i^2$$

if one of the alternating edge conditions is fully satisfied on the subset $E_{odd} = \{e_1, e_3, ..., e_{2m-1}\}$ consisting of all edges with odd indices.

Proof. We begin by showing that $\mathrm{L} := (\ell_1, 0, \ell_3, 0, ..., \ell_{n-1}, 0) \in \Delta_\lambda$ (i.e. that L is a valid packing). First, we assess the backward alternating edge condition. By definition, we have that $\lambda_i + \lambda_{i+1} \leq \ell_i$ and $\lambda_{i+1} + \lambda_{i+2} \leq \ell_{i+1}$. Setting $\lambda_i = \ell_i$ and $\lambda_{i+1} = 0$, for all odd $0 < i < n$, we observe that $\lambda_i + \lambda_{i+1} = \ell_i + 0 = \ell_i$, and $\lambda_{i+1} + \lambda_{i+2} = 0 + \ell_{i+2} = \ell_{i+2}$, which, by the backward alternating edge condition, is less than or equal to ℓ_{i+1}. The argument for forward alternating edge condition is similar. Therefore, we have $\mathrm{L} \in \Delta_\lambda$. But we also have

$$\begin{aligned}
\frac{1}{2}||\mathbf{\Lambda}||_2^2 &= \frac{1}{2}\sum_{i=1}^{n}\lambda_i^2 \\
&= \frac{1}{2}\sum_{i \text{ odd}}^{n}(\lambda_i^2 + \lambda_{i+1}^2) \\
&\leq \frac{1}{2}\sum_{i \text{ odd}}^{n}(\lambda_i + \lambda_{i+1})^2 \\
&\leq \frac{1}{2}\sum_{i \text{ odd}}^{n}\ell_i^2
\end{aligned} \tag{3}$$

such that $\frac{1}{2}\sum\limits_{i \text{ odd}}^{n}\ell_i^2$ is an upper bound on the packing area. Note that Eq 3 uses the fact that the 2-norm is bounded above by the 1-norm. Now, we set $\mathrm{L} = \mathbf{\Lambda}$, and $\mathbf{L}$ is our maximum packing area.

■

Remark 3.10. By re-indexing, we can generalize this result to any polygon that satisfies one of the alternating edge conditions on alternating edges. In addition, notice that this maximum packing is non-unique, as both the forward and backwards conditions can be achieved simultaneously, as in the case of a rational square.

Theorem 3.11. *The 2-dimensional packing problem that fully satisfies the alternating edge condition can be solved in $\mathcal{O}(n)$ steps*

Proof. By lemma 3.9, check that the vertices of the polygon satisfy the condition; this can be done in $\mathcal{O}(n)$ steps. ■

4 Algorithm

In this section we explain the algorithm we use to compute the packing capacity of a given polygon, making use of the theoretical results above.
Overall expected Input: Vertices Coordinate of Polygon in counterclockwise direction
Overall desired Output: 1.Size of max packing area 2.Value of $\lambda_1 ... \lambda_n$ 3.Packing visualization

4.1 Scale polygon Algorithm

Expected Input: Vertices Coordinate of Polygon in rational format
Desired Output: Vertices Coordinate of Polygon in integer format

Construct two arrays that contains the numerators and denominators of the vertices, respectively. We then multiply the vertices with the least common multiple of the denominators to remove the fractional form. The resulting coordinates of the vertices are integers.

4.2 $\mathrm{SL}_2(\mathbb{Z})$ Algorithm

Expected Input: Coordinates of two adjacent vertices
Desired Output: The $\mathrm{SL}_2(\mathbb{Z})$ length between those two vertices
Given the integer coordinates of two vertices $v_1 = (a_1, b_1), v_2 = (a_2, b_2)$, by Theorem 2.13, the $\mathrm{SL}_2(\mathbb{Z})$ length between these two vertices is the greatest common divisor of $|b_2 - b_1|$ and $|a_2 - a_1|$.

4.3 Get Candidate Algorithm

Expected Input: List of $\mathrm{SL}_2(\mathbb{Z})$
Desired Output: All possible packings solved from the permuted matrix.
Given the Constrains: $\lambda_i + \lambda_{i+1} \leq \ell_i$ and $\lambda_i \geq 0$, there are $2n$ inequalities. We can build two matrix equation by turning those $2n$ inequalities into equalities. The first linear system matrix, using the first n inequalities gives:

$$\begin{pmatrix} 1 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{pmatrix} = \begin{pmatrix} \ell_1 \\ \ell_2 \\ \vdots \\ \ell_n \end{pmatrix}$$

The second linear system matrix, using the last n inequalities gives:

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

In total, there are $\binom{2n}{n}$ rows we can choose to make a linear system matrix. For example, those 2 rows in first matrix equation and $n-2$ rows in second matrix equation and we have:

$$\begin{pmatrix} 1 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{pmatrix} = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ 0 \end{pmatrix}$$

If there is one unique solution for λ under this specific matrix equations. We will keep the value of this set of λ for possible candidates.

4.4 λ Filter Algorithm

Expected Input: Values of $SL_2(\mathbb{Z})$
Desired Output: List of all feasible packings by the non-decreasing order of packing area magnitude
Magnitude packing is calculated by the formula:

$$\frac{1}{2}\sum_{i=1}^{n}\lambda_i^2$$

For each candidate packing, check whether it satisfies all $2n$ inequalities in Section 4.3 and then sort all feasible packings with the non-decreasing order of its magnitude.

4.5 Graph Visualization Algorithm

Expected Input: Vertices Coordinate of Polygon by counterclockwise direction, lengths of $SL_2(\mathbb{Z})$ and current values of λ.
Desired Output: Visualization that shows original polygon and packing area.
To draw the triangular packing area, the vertex coordinates of packing area are indispensable. When the given value of λ_i and ℓ_i, the ratio of them is easy to get. Suppose we focus on the vertex, v_i and its adjacent vertices are v_{i-1},v_{i+1} and $n = \frac{\lambda_{i-1}}{\ell_{i-1}}$ and $m = \frac{\lambda_{i+1}}{\ell_{i+1}}$. The vertices coordinate of packing area in v_i are

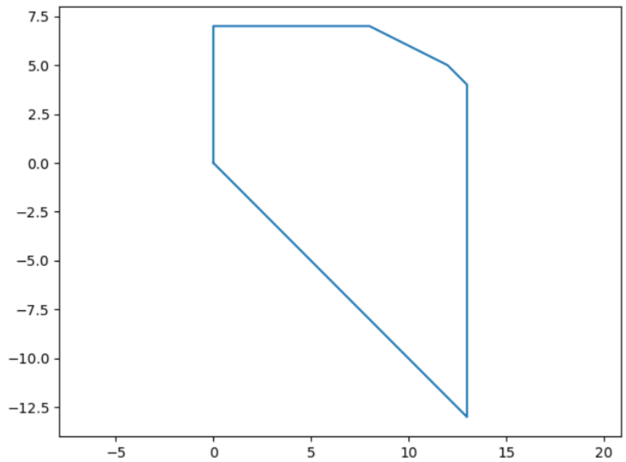
$$v_i; v_i + (v_{i-1} - v_i) \times n; v_i + (v_{i+1} - v_i) \times m \tag{4}$$

With all necessary vertex coordinates, program takes the advantage of Python library called matplotlib to draw two-layer graph.

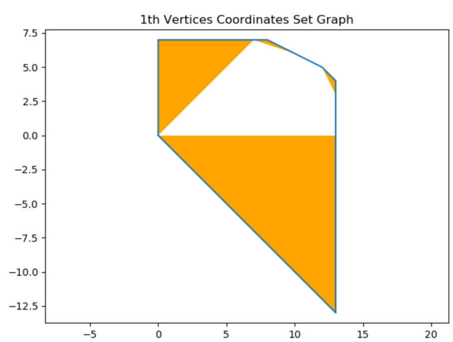
5 Examples

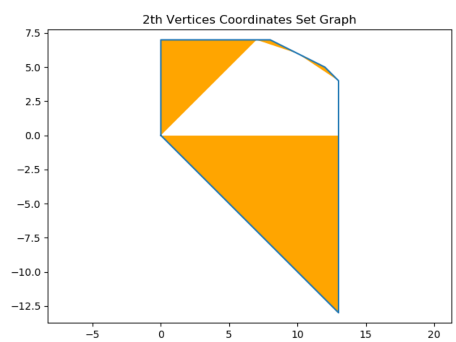
5.1 Detailed Demo

Suppose we have a polygon with vertices coordinates: (0,0); (13,-13); (13,4); (12,5); (8,7); (0,7). The polygon shows below



Step1: Check whether scale function needed. The answer is No.
Step2: The program outputs the $SL_2(\mathbb{Z})$: [13, 17, 1, 2, 8, 7]
Step3: Get all packing conditions and output λ values with maximum packing area: [0.0, 13.0, 1.0, 0.0, 1.0, 7.0] and [0.0, 13.0, 0.0, 1.0, 1.0, 7.0]
Step4: Get vertices coordinate of each triangular packing area and plot in the graph:



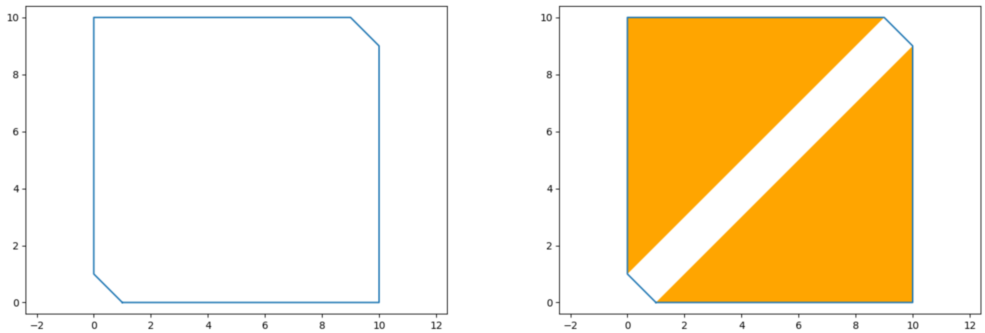


Notice: All four steps would run automatically and program output all information and generate graph in one step. Program will display all packing conditions that share the same maximum magnitude if second parameter of get_info function is True.

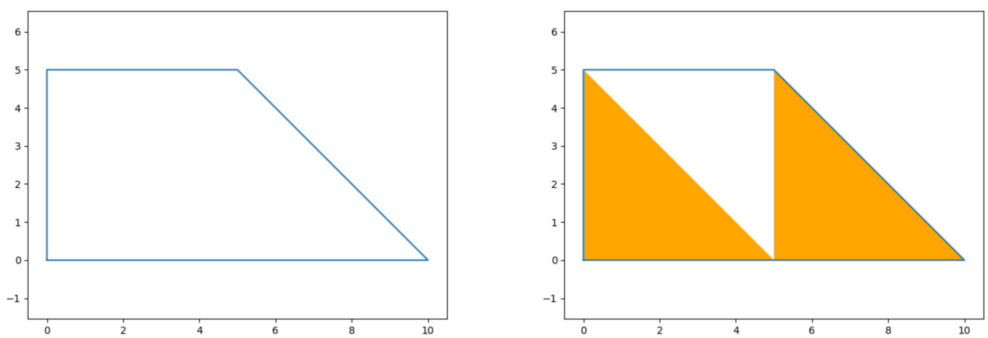
5.2 More Demo Graphs

Input the polygon: (1,0); (10,0); (10,9); (9,10); (0,10); (0,1)
Output:
1. $SL_2(\mathbb{Z})$: [9, 9, 1, 9, 9, 1]
2. λ values: [0.0, 9.0, 0.0, 0.0, 9.0, 0.0]

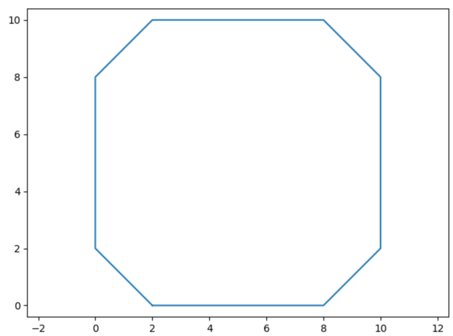
3. Packing magnitude: 81
Polygon Visualization below:

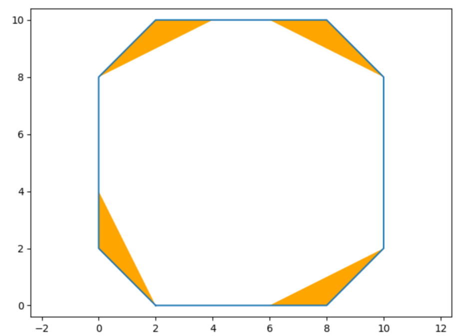


Input the polygon: (0,0); (10,0); (5,5); (0,5)
Output:
1. $SL_2(\mathbb{Z})$: [10, 5, 5, 5]
2. λ values: [5.0, 5.0, 0.0, 0.0]
3. Packing magnitude: 25
Polygon Visualization below:



Input the polygon: (2,0); (8,0); (10,2); (10,8); (8,10); (2,10); (0,8); (0,2)
Output:
1. $SL_2(\mathbb{Z})$: [6, 2, 6, 2, 6, 2, 6, 2]
2. λ values: [0.0, 2.0, 0.0, 0.0, 2.0, 2.0, 0.0, 2.0]
3. Packing magnitude: 8
Polygon Visualization below:





6 Conclusion

In this project, we proved two theorems that allow us to derive an algorithm that maximizes the packing area explicitly. With these tools, we developed a Python script that computes the packing area of Delzant polygons exactly. There are many future directions in which this work can extend. For the Delzant case, the study focus can be both theoretical and practical.

- We would like to determine if there is an analogous result to the even alternating edge case for Delzant polygons with an odd number of edges.
- Further study can also be done on the behavior of an operation called corner chops on the packings of Delzant polygons.
- Regarding the algorithm, we can add more features to the program such as
  - an interactive graphing tool where a user can draw an input polygon
  - handling of irrational vertices, performing corner chops on the input polygon
  - implementing new technologies to decrease time complexity

In addition, we are interested in studying the packing capacities of the semitoric polygons. Works regarding the semitoric case include

- discovering any interesting patterns or methods pertaining to the packing of semitoric polygons.
- adapting the current algorithm to compute the maximum valid packings of semitoric polygons.

7 Appendix

Source Code Github Link:
https://github.com/CoulsonZhang/Semi-toric_Packing_Capacity

8 References

[1] A. Figalli, J. Palmer, and Á. Pelayo. Symplectic G-capacities and integrable systems. *Ann. Sc. Norm. Super. Pisa Cl. Sci. (5)*, 18(1):65–103, 2018.

[2] Á Pelayo, Toric symplectic ball packing, *Topology and its Appl. 157* (2006), 3633–3644.