2 Non-Simple Polygons

2.1 Let me not be pent up, sir; I will fast, being loose.

*Fast and Loose* is the name of a family of magic tricks (or con games) with ropes, chains, and belts that have been practiced since at least the 14th century; the con game is mentioned in three different Shakespeare plays. In one such trick (now sometimes called the *Endless Chain*), the con artist arranges a closed loop of chain into a figure-8, and then asks the mark to put their finger on the table inside one of the loops. The con artist them pulls the chain along the table. If the chain catches on the mark's finger, then the chain is *fast* and the mark wins; if the con artist can pull the chain completely off the table, the chain is *loose* and the mark loses.

The con artist shows the mark that there are two different ways for the loops to fall. (Notice how the chain crosses itself in the lower corners.) Because the chain is bright and shiny and bumpy, it's impossible for the mark to tell which way the chain is actually arranged, but because these are the only possibilities, the mark should have a 50-50 chance of winning. Right? Right?

![Figure 1: Two arrangements of the Endless Chain](image1.png)

Of course not! As soon as the mark places money on the barrelhead, the con artist wins every time. The con artists was lying; there is a third arrangement of the chain that is *always* loose, no matter where the mark puts their finger.

![Figure 2: The actual arrangement of the Endless Chain](image2.png)

2.2 Shoelaces and Signed Areas

[Write this!]

```python
def signedArea(P):
    area = 0
    n = size(P)
    for i in range(n):
        area += (P[i].x * P[(i+1)%n].y - P[i].y * P[(i+1)%n].x) / 2
    return area
```


2.3 Winding numbers

The winding number of a polygon $P$ around a point $o$ is intuitively (and not surprisingly) the number of times that $P$ winds counterclockwise around $o$. For example, if $P$ is a simple polygon, its winding number around any exterior point is zero, and its winding number around any interior point is either $+1$ or $-1$, depending on how the polygon is oriented. If the polygon winds clockwise around $o$, the winding number is negative. Crucially, the winding number is only well defined if the polygon does not contain the point $o$.

We can define the winding number more precisely in terms of angles as follows. Let $p_0, p_1, \ldots, p_{n-1}$ denote the vertices of $P$ in order. For each index $i$, let $\theta_i$ denote the interior angle at $o$ in the triangle $\triangle p_i o p_{i+1}$, with positive sign if $(0, p_i, p_{i+1})$ is oriented counterclockwise, and with negative sign if $(0, p_i, p_{i+1})$ is oriented clockwise. Assuming angles are measured in circles (the way the gods intended, as opposed to radians or degrees or some other idiocy), the winding number of $P$ around $o$ is the sum $\sum_i \theta_i$.

Actually computing the winding number according to this definition requires inverse trigonometric functions, square roots, and other numerical madness. Fortunately, there is an equivalent definition that builds on our ray-shooting test from the previous lecture. Let $R$ be a vertical ray shooting upward from $o$. We distinguish two types of crossings between the $R$ and the polygon, depending on the orientation of the crossed edges. Specifically, if the crossed edge is directed from right to left, we have a positive crossing; otherwise, we have a negative crossing.
Equivalently, when $R$ crosses an edge $p_ip_{i+1}$, the sign of the crossing is the sign of the determinant $\Delta(o, p_i, p_{i+1})$.

Figure 5: A positive crossing (left) and a negative crossing (right)

I'll leave the equivalence of these two definitions as an exercise. (Hint: prove equivalence for triangles, and then look at Meister's figure again!)

Here is the ray-shooting algorithm in (pseudo)Python. Any similarities with the point-in-polygon algorithm from the previous lecture are purely intentional.

```python
def windingNumber(P, o):
    wind = 0
    n = size(P)
    for i in range(n):
        p = P[i]
        q = P[(i+1)%n]
        Delta = (p.x - o.x)*(q.y - o.y) - (p.y - o.y)*(q.x - o.x)
        if p.x <= o.x < q.x && Delta > 0:
            wind += 1
        elif q.x <= o.x < p.x && Delta < 0:
            wind -= 1
    return wind
```

Figure 6: Winding numbers of the Endless Chain around various points

2.4 Homotopy

A homotopy between two closed curves is a continuous deformation—a morph—from one curve to the other. Homotopies can be defined between curves in any topological space, but for purposes of illustration, let's restrict ourselves to curves in the punctured plane $\mathbb{R}^2 \setminus o$, where $o$ is an arbitrary point called the obstacle.

Formally, a free homotopy between two closed curves in $\mathbb{R}^2 \setminus o$ is a continuous function $h : [0, 1] \times S^1 \rightarrow \mathbb{R}^2 \setminus o$, such that $h(0, \cdot)$ and $h(1, \cdot)$ are the initial and final closed curves, respectively.
each $0 < t < 1$, the function $h(t, \cdot)$ is the intermediate closed curve at “time” $t$. Crucially, none of these intermediate curves touches the obstacle point $o$.

We also need a definition of homotopy between paths; this is a little more subtle. Let $\pi : [0, 1] \to \mathbb{R}^2 \setminus o$ and $\sigma : [0, 1] \to \mathbb{R}^2 \setminus o$ be two paths in the punctured plane the same endpoints: $\pi(0) = \sigma(0)$ and $\pi(1) = \sigma(1)$. A path homotopy from $\pi$ to $\sigma$ is a continuous function $h : [0, 1] \times [0, 1] \to \mathbb{R}^2 \setminus o$ that satisfies four conditions:

- $H(0, t) = \pi(t)$ for all $t$
- $H(1, t) = \sigma(t)$ for all $t$
- $H(s, 0) = \pi(0) = \sigma(0)$ for all $s$
- $H(s, 1) = \pi(1) = \sigma(1)$ for all $s$

Intuitively, you should think of a homotopy as a continuous deformation of one path into the other, keeping the endpoints fixed at all times. Again, for each $0 < s < 1$, the function $h(t, \cdot)$ is the intermediate path at “time” $s$, and none of these intermediate paths touches the obstacle point $o$.

I’ll typically use the word “homotopy” for both types of functions, in the hope that the precise type is clear from context.

Two closed curves in $\mathbb{R}^2 \setminus o$ are homotopic, or in the same homotopy class, if there is a homotopy from one to the other in $\mathbb{R}^2 \setminus o$. Homotopy is an equivalence relation.

A closed curve is contractible in $\mathbb{R}^2 \setminus o$ if it is homotopic to a single point (or more formally, to a constant curve).

### 2.5 Vertex moves

Similar to the definition of “connected”, the definition of “homotopy” allows intermediate curves to be arbitrarily wild closed curves even if the initial and final curves are polygons.

Fortunately, there is a general principle that allows us to “tame” homotopies between tame curves like polygons, by decomposing them into a sequence of elementary moves. (This principle is similar to the observation that any closed curve can be approximated by a sequence of line segments, otherwise known as a polygon.)

Let $P$ be any polygon. A vertex move translates exactly one point $p$ of $P$ along a straight line from its current location to a new location $p'$, yielding a new polygon $P'$. You should imagine that as the point $p$ moves, the edges incident to $p$ pivot around their other endpoints. Typically the moving point $p$ is a vertex of the initial polygon $P$ and the final point $p'$ is a vertex of the final polygon $P'$, but neither of these restrictions is required by the definition. We are allowed to introduce new vertices in the middle of edges, or to delete “flat” vertices between two collinear edges, at will.

**Figure!**

Now suppose the polygon $P$ lives in the punctured place $\mathbb{R}^2 \setminus o$. Let $p, q, r$ be three consecutive vertices of $P$. The vertex move $q \to q'$ is safe if neither of the triangles $\triangle pqq'$ or $\triangle qq'r$. contains the obstacle point $o$. Equivalently, during a safe vertex move, the continuously changing polygon never touches $o$.

It follows that every safe vertex move is a homotopy in $\mathbb{R}^2 \setminus o$. We can build up more complex
homotopies by concatenating several safe vertex moves. In fact, any sequence of safe vertex moves describes a homotopy in $\mathbb{R}^2 \setminus o$.

2.6 Polygon homotopies are sequences of vertex moves

Unfortunately, the converse of this observation is false; not every homotopy is a sequence of vertex moves. Consider, for example, a simple translation or rotation of the entire polygon! Nevertheless, every homotopy can be approximated by a sequence of safe vertex moves.

Lemmas: If two polygons in $\mathbb{R}^2 \setminus o$ are homotopic, then they are homotopic by a sequence of safe vertex moves.

Proof: Fix a homotopy $h : [0, 1] \times S^1 \to \mathbb{R}^2 \setminus o$ between two polygons $P_0 = h(0, \cdot)$ and $P_1 = h(1, \cdot)$.

For any parameters $t$ and $\theta$, let $d(t, \theta)$ be the Euclidean distance from $h(t, \theta)$ to the origin $o$, and let $\varepsilon = \min_{t, \theta} d(t, \theta)$. Because the cylinder $[0, 1] \times S^1$ is compact, this minimum is well-defined and positive.

We subdivide the cylinder $[0, 1] \times S^1$ into triangles as follows. First, cut the cylinder into a grid of $\delta \times \delta$ squares $\Box(i, j) = [i\delta, (i + 1)\delta] \times [j\delta, (j + 1)\delta \mod 1]$, where $\delta > 0$ is chosen so that the diameter of $h(\Box(i, j))$ is at most $\varepsilon/2$. (The existence of $\delta$ is guaranteed by continuity. Then further subdivide each grid square into two right isosceles triangles. Without loss of generality, assume each vertex of $P_0$ and $P_1$ the image of some vertex on the boundary of the resulting triangle mesh $\Delta$.

Figure!!

We can easily construct a sequence of $1 + 2/\delta^2$ cycles in $\Delta$ that starts with one boundary $0 \times S^1$ and ends with the other boundary $1 \times S^1$, such that the symmetric difference between two adjacent cycles is the boundary of one triangle in $\Delta$. The homotopy $h$ maps any two adjacent cycles in this sequence to a pair of polygons that differ by a triangle move.

Thus, we obtain a sequence of $1 + 2/\delta^2$ vertex moves transforming $P_0$ into $P_1$. Every vertex of each intermediate polygon has distance at least $\varepsilon$ from the origin, each edge has length at most $\varepsilon/2$, and each vertex move translates its vertex a distance of at most $\varepsilon/2$. It follows that every vertex move in this sequence is safe.

This observation is a special case of a more general simplicial approximation theorem, which intuitively states that any continuous map between nice topological spaces can be approximated by a nice continuous map; moreover, the original map and its approximation are homotopic.

2.7 Homotopy Invariant

Winding numbers are our first example of a topological invariant, and specifically a complete homotopy invariant. A topological invariant is any property of objects or spaces that is unchanged by homeomorphism; a standard example for connected orientable surfaces is the genus. A homotopy invariant is any property that is preserved by homotopy; a homotopy invariant is complete if it takes on different values for two objects that are not homotopic.

Theorem: Two polygons are homotopic in $\mathbb{R}^2 \setminus o$ if and only if they have the same winding number around the origin $o$. 
**Proof:** Fix two polygons $P_0$ and $P_1$ in $\mathbb{R}^2 \setminus o$. If these two polygons are homotopic, then by the previous lemma, they are connected by a sequence of safe triangle moves. A safe triangle move does not change the winding number of a polygon around the origin. Thus, by induction, $P_0$ and $P_1$ have the same winding number.

To prove the converse, I'll describe a sequence of safe triangle moves that transforms any polygon $P$ into a canonical polygon $\diamond^w$ with the same winding number $w$ around the origin. (The notation $\diamond^w$ will make sense later, honest.) Thus, if $P_0$ and $P_1$ have the same winding number $w$, we can deform $P_0$ into $P_1$ by concatenating the move sequence that takes $P_0$ to $\diamond^w$ and the reverse of the move sequence that takes $P_1$ to $\diamond^w$.

Our homotopy consists of several stages. First let's consider the case where the winding number of $P$ around 0 is not zero.

- Let $p_i$ be any vertex of $P$, and let $p_{i-1}$ and $p_{i+1}$ be the next and previous vertices. We call $q$ redundant if the triangle $\triangle p_{i-1}p_ip_{i+1}$ does not contain the origin. In particular, if the triples $(o, p_{i-1}, p_i)$ and $(o, p_i, p_{i+1})$ have opposite orientations, one clockwise and the other counterclockwise, then $p_i$ is redundant. In the first phase of our homotopy, we repeatedly remove redundant vertices, by moving each redundant vertex $q$ to one of its neighbors, until none are left. The resulting polygon $P'$ is angularly monotone: every triple $(o, p_i, p_{i+1})$ has the same orientation.

- Next, we subdivide $P'$ by adding vertices at its intersections with rays pointing up, down, left, and right from the origin $o$. After this subdivision, any vertex that is not on one of these rays is redundant. So in the second phase of the homotopy, we remove all non-ray vertices using safe vertex moves. The resulting polygon $P''$ is still angularly monotone.

- Finally, we move each vertex so that its distance from the origin is 1; each of these vertex moves is safe. The resulting polygon $\diamond^w$ has vertices only at the points $(0, 1), (1, 0), (0, -1), \text{ and } (-1, 0)$; the polygon winds around this diamond $|w|$ times, counterclockwise if $w > 0$ and clockwise if $w < 0$.

The special case where $P$ has winding number 0 is even simpler. The first phase (removing redundant vertices) actually reduces $P$ to a single point; we can then translate this point to $\diamond^0 = (1, 0)$ using one more safe vertex move.

**Figures!!**

This theorem immediately implies a linear-time algorithm to decide if two polygons are homotopic in the punctured plane: Count how many times each polygon crosses an arbitrary ray from the origin in each direction.

*(This is as far as I'll get on Thursday)*

---

### 2.8 Multiple Obstacles

Now suppose that there is more than one obstacle point. Let $O = \{a, b, c, \ldots\}$ be an arbitrary finite set of points in the plane. The definition of homotopy easily generalizes to polygons (and other closed curves) in $\mathbb{R}^2 \setminus O$. How quickly can we tell whether two polygons in $\mathbb{R}^2 \setminus O$ are homotopic?
The game Fast and Loose shows some of the subtlety of this problem. Let $a$ and $b$ be arbitrarily points in each of the two loops of the curve $C$ that the con artist actually uses. It’s not hard to see that $\text{wind}(c, a) = \text{wind}(C, b) = 0$. Thus, the curve is contractible in the plane with only one of these punctures; in other words, the chain is loose if we only use one finger. But the curve $C$ is not contractible in $\mathbb{R}^2 \setminus \{a, b\}$; we can hold the chain fast by placing a finger in each loop. Winding numbers are not a homotopy invariant when there is more than one obstacle.

However, we can still define a homotopy invariant by shooting rays out of every obstacle. Assume without loss of generality that the obstacles have distinct $x$-coordinates. Shoot a vertical ray upward from each obstacle point. The crossing sequence of a polygon $P$ in $\mathbb{R}^2 \setminus O$ is the sequence of intersections between these vertical rays and the polygon, in order along the polygon, along with the sign of each crossing (positive if the polygon crosses the ray to the left, negative if the polygon crosses the ray to the right).

The figure below shows a polygon in the plane with two obstacle points $a$ and $b$. If we orient the polygon as indicated by the arrows, starting at the lower left corner, the crossing sequence is $\text{BAabBAabB}$, where each upper-case letter denotes a positive crossing through the corresponding ray, and each lower-case letter denotes a negative crossing through the corresponding ray.

### 2.9 Reductions

We regard signed crossing sequences as strings of abstract symbols, where each symbol $a$ has a formal “inverse” $\bar{a}$. In our earlier example, each upper case letter is the inverse of the corresponding lower-case letter, and vice versa. Let $x \cdot y$ denote the concatenation of strings $x$ and $y$, and let $\epsilon$ denote the empty string.

An elementary reduction is a transformation of the form $x \cdot a\bar{a} \cdot y \mapsto x \cdot y$, where $x$ and $y$
are (possibly empty) strings and $a$ is a single symbol. An elementary expansion is the reverse transformation $x \cdot y \to x \cdot a \cdot y$. Two strings are equivalent if once can be transformed into the other by a sequence of elementary reductions and expansions. (Check for yourself that equivalence is in fat an equivalence relation!) We call a string trivial if it is equivalent to the empty string $\epsilon$. Finally, a string is reduced if no elementary reductions are possible; for example, the empty string $\epsilon$ is trivially reduced, as is any string of length 1.

Crossing sequences of polygons are actually cyclic strings. Formally, a cyclic string is an equivalence class of linear strings:

$$(w) := \{y \cdot x \mid x \cdot y = w\}$$

For example, $(ABbA) = \{ABa, BbA, bAB, AAb\}$ and $(\epsilon) = \{\epsilon\}$. I’ll write $w \sim z$ to denote that $w$ is a cyclic shift of $z$, or equivalently $w \in (z)$, or equivalently $z \in (w)$. To emphasize that elementary reductions can “wrap around” cyclic strings, we say that a cyclic string is cyclically reduced if no elementary reductions are possible. A (cyclic) string is trivial if it is equivalent to the empty (cyclic) string.

For example, the cyclic string $(EcaCbaAbcEeEeAdbcCBaEdDeADCe)$ is trivial; two different sequences of elementary reductions are shown below (using interpuncts to represent missing symbols). In the first sequence, each elementary reduction removes the leftmost matching pair; the second sequence is more haphazard. In fact, as the following lemma implies, any sequence of elementary reductions eventually reduces this string to nothing.

```
EcaCbaAbcEeEeAdbcCBaEdDeADCe
EcaCb · BcEeEeAdbcCBaEdDeADCe
EcaC · · · cEeEeAdbcCBaEdDeADCe
Eca · · · · · · EeEeAdbcCBaEdDeADCe
Eca · · · · · · · AdbcCBaEdDeADCe
Ec · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · }
We finally have all the ingredients of our homotopy-testing algorithm.

**Proof (sketch):**
Every homotopy can be approximated by a sequence of safe vertex moves, and each safe vertex move changes the signed crossing sequence by an explicit sequence of elementary reductions and their inverses. Conversely, any elementary reduction of the signed crossing sequence can be modeled by a sequence of safe vertex moves. Figures!

We finally have all the ingredients of our homotopy-testing algorithm.

**Theorem:** For any set $O$ of $k$ points in $\mathbb{R}^2$, and any two $n$-gons $P$ and $Q$ in $\mathbb{R}^2 \setminus O$, we can determine whether $P$ and $Q$ are homotopic in $\mathbb{R}^2 \setminus O$ in $O(k \log k + kn)$ time.

**Proof (sketch):** As usual we assume without loss of generality that the obstacles and polygon vertices all have distinct $x$-coordinates. First we sort the obstacles from left to right in

---

**Lemma:** Every cyclic string is equivalent to exactly one cyclically reduced cyclic string.

**Proof:** Let $w$ be a cyclic string that allows two elementary reductions $w \leftrightarrow x$ and $w \leftrightarrow y$, meaning two different pairs of symbols are deleted. We claim that either $x = y$ or there is another string $z$ such that $x \leftrightarrow z$ and $y \leftrightarrow z$ are elementary reductions.

- If the pairs of symbols deleted by $w \leftrightarrow x$ and $w \leftrightarrow y$ are disjoint, then we can write $w = (a_1 \cdot w_1 \cdot c_1 \cdot w_2)$ for some (possibly empty) linear strings $w_1$ and $w_2$ and (possibly equal, possibly inverse) symbols $a$ and $c$. Without loss of generality we have $x = (w_1 \cdot c \cdot w_2)$ and $y = (w_1 \cdot a \cdot w_2)$. In this case, we can take $z = (w_1 w_2)$.

- If the pairs of symbols deleted by $w \leftrightarrow x$ and $w \leftrightarrow y$ overlap, then we can write $w = (a \cdot w')$ for some (possibly empty) linear string $w'$ and some symbol $a$. In this case we have $x = y = (a \cdot w')$.

It follows that applying only elementary reductions leads to a unique reduced string; however, equivalence also allows elementary expansions. Consider two equivalent but distinct cyclic strings $x \neq y$, and let $x = w_1 \leftrightarrow w_2 \leftrightarrow \cdots \leftrightarrow w_n = y$ be a sequence of strings, each connected to its success by an elementary reduction in one direction $w_i \leftrightarrow w_{i+1}$ or the other $w_{i+1} \leftrightarrow w_i$.

Suppose for some index $i$, we have reductions $w_i \leftrightarrow w_{i-1}$ and $w_j \leftrightarrow w_{j+1}$. If $w_{i-1} = w_{j+1}$, then we can remove $w_{i-1}$ and $w_j$ to obtain a shorter transformation sequence. Otherwise, there is another string $z_i$ such that $w_{i-1} \leftrightarrow z_i$ and $w_{i+1} \leftrightarrow z_{i+1}$. Thus, by induction, we can modify our transformation sequence so that every reduction appears before every expansion.

Let $z$ be the shortest string in this normalized sequence. Both $x$ and $y$ can be reduced to $z$ using only elementary reductions. Because $x \neq y$, either $x \neq z$ or $y \neq z$; we conclude that at most one of $x$ and $y$ is reduced.

**Lemma:** Any cyclic string of length $x$ can be cyclically reduced in $O(x)$ time.

**Proof (sketch):** (It’s the three-penny algorithm again.) Details!
$O(k \log k)$ time. Then we compute the signed crossing sequence of $P$ and $Q$, each in $O(nk)$ time; each signed crossing sequence has length $O(nk)$. Then we cyclically reduce the two crossing sequences in $O(nk)$ time. Finally, we check whether the two reduced crossing sequences are equal (as cyclic strings) in linear time using any fast string-matching algorithm.

2.10  ...and the Aptly Named Sir Not Appearing in This Film

- alternativee fences for crossing sequences
- rotation number = total turning angle = smiles − frowns
- regular homotopy = vertex moves without spurs
- rotation number is a regular homotopy invariant
- complex root finding
- picture hanging puzzles
- signed volumes of self-intersecting polyhedra (hic utres unilateralis nascuntur)