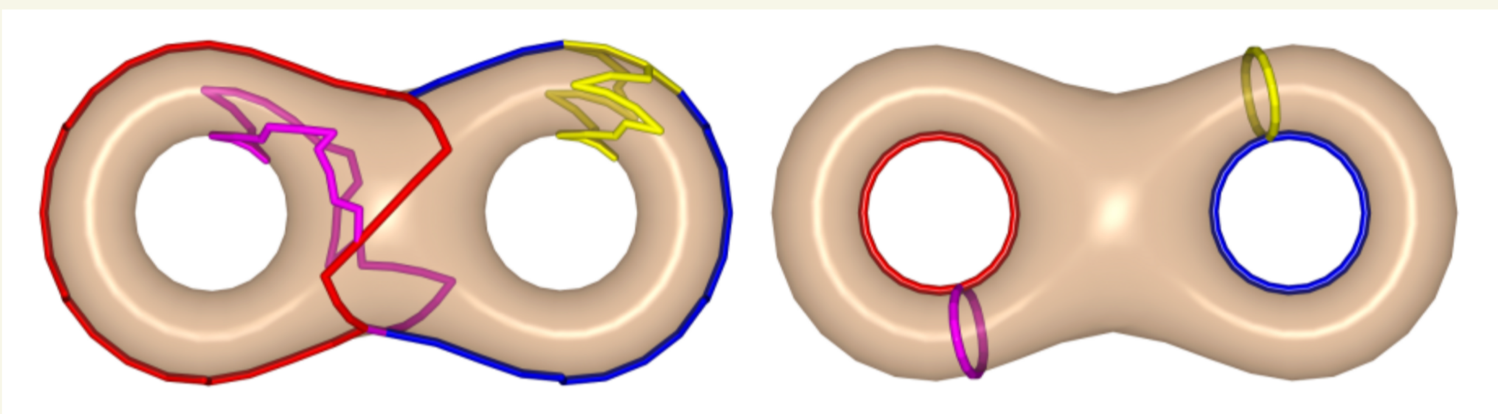
TDA - fall 2025

Optimal
Cycles

Homology generators

Many different cycles exist in a homology class, but computing a cycle basis can have interesting applications.



Issue: What is optimal?

Given a measure on complex K, often want minimum representatives

Let $w: K_p \to \mathbb{R}_{\geq 0}$ be a non-negative weight function on p-simplices of K.

Given a cycle c ($\mathbb{Z}_2$-homology, so $c = \sum_{\text{simplices } \sigma_i} (\alpha_i \cdot \sigma_i)$, $\alpha_i \in \{0,1\}$), the weight of c, $w(c) = \sum_i \alpha_i \cdot w(\sigma_i)$

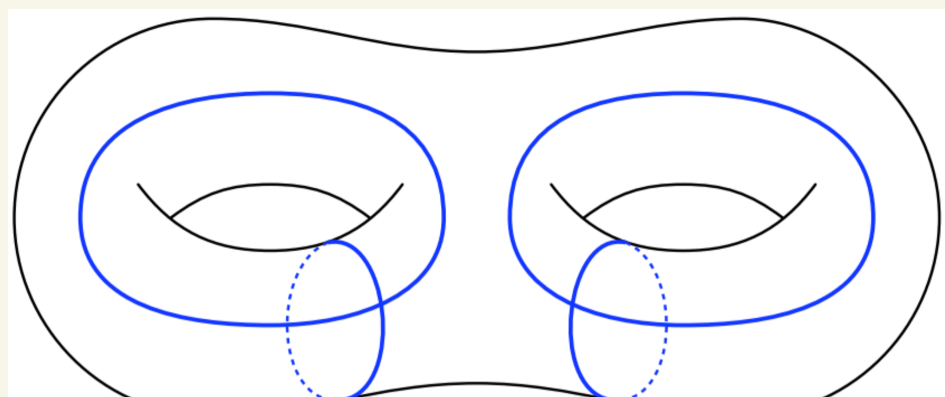
For a set of cycles $C = \{c_1, \dots, c_g, \; c_i \in Z_p(K)\}$

$$w(C) = \sum_{i=1}^{g} w(c_i).$$

Optimality

We say a set of cycles $C = \{c_1, ..., c_g\}$ is an H_p-basis is $\{[c_i], i = 1, .., d\}$ generate $H_p(K)$ and $d = \dim(H_p(K))$, + C is optimal if there is no other generator C' with $w(C') < w(C)$

Optimal Homology Basis Problem (OHBP): Find the best such basis.

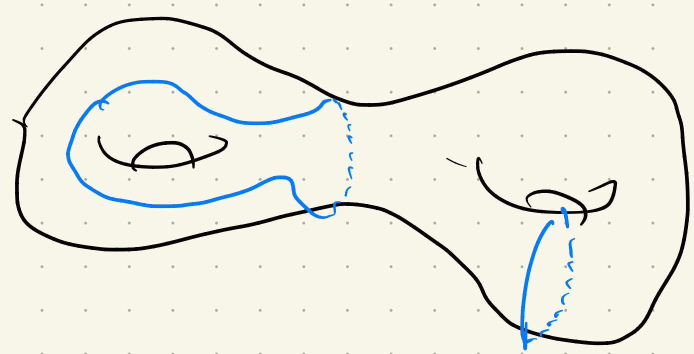


(d)

[Erickson & Whittlesey 2005]

An algorithm for H_1:

We say a cycle is tight if it contains the shortest path between all points on the cycle.

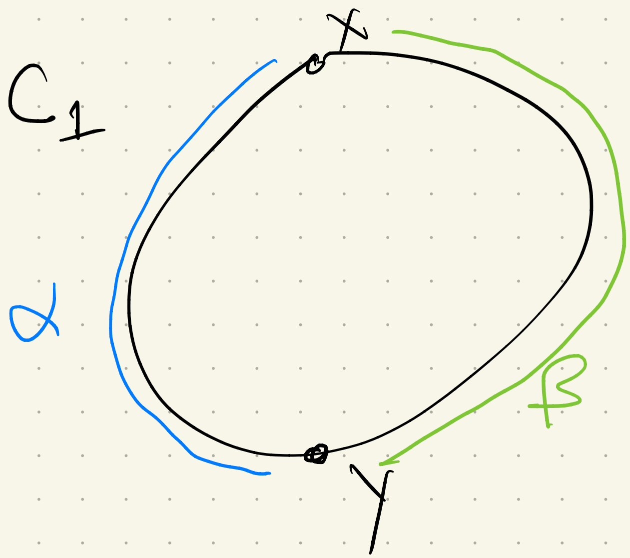


Claim: Any cycle in optimal homology basis is tight.

Proof: Spps not: $c_1, \ldots, c_{2g}$ but c_1 is not tight

Could be disconnected:

or, not all shortest paths:



Pick 2 points on cycle such that C_1 doesn't contain x-y shortest path.

Consider shortest path σ:

Shortest path trees

Dijkstra's algoritm takes a source vertex s in a graph & computes the set of all minimum s-v paths

```
function Dijkstra(Graph, source):

    for each vertex v in Graph.Vertices:
        dist[v] ← INFINITY
        prev[v] ← UNDEFINED
        add v to Q
    dist[source] ← 0

    while Q is not empty:
        u ← vertex in Q with minimum dist[u]
        Q.remove(u)

        for each arc (u, v) in Q:
            alt ← dist[u] + Graph.Edges(u, v)
            if alt < dist[v]:
                dist[v] ← alt
                prev[v] ← u

    return dist[], prev[]
```

(actually Leyzorek et al '57, Dantzig '58)

Classic example of a "greedy algorithm".

Runtime:

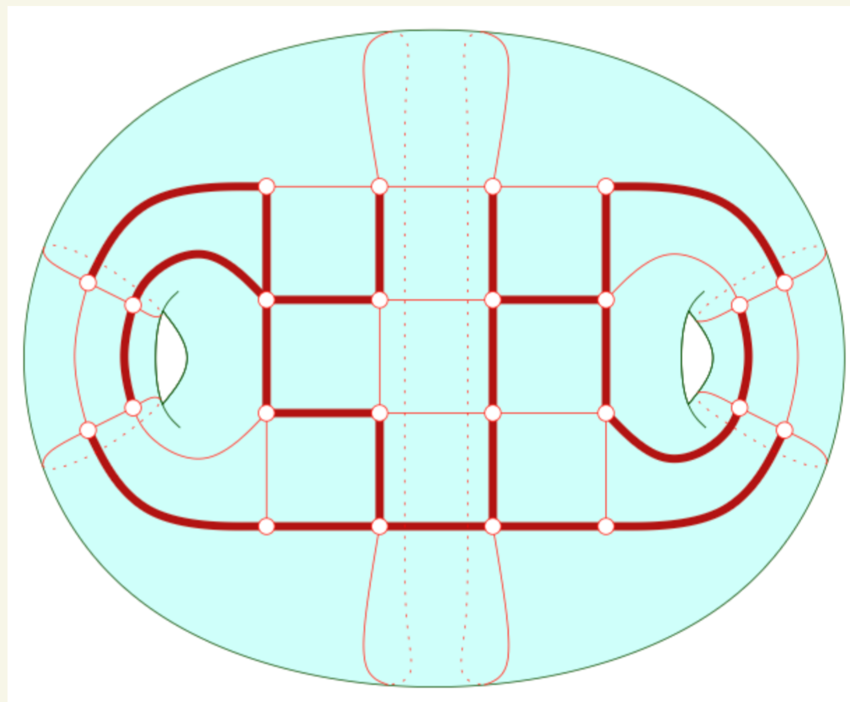
(show demo)

Back to cycles

We can use shortest path trees to generate candidate cycles for OHBP.

How?

Consider such a tree: need cycles where $\forall u, v$, shortest path in cycle:



Book's version:

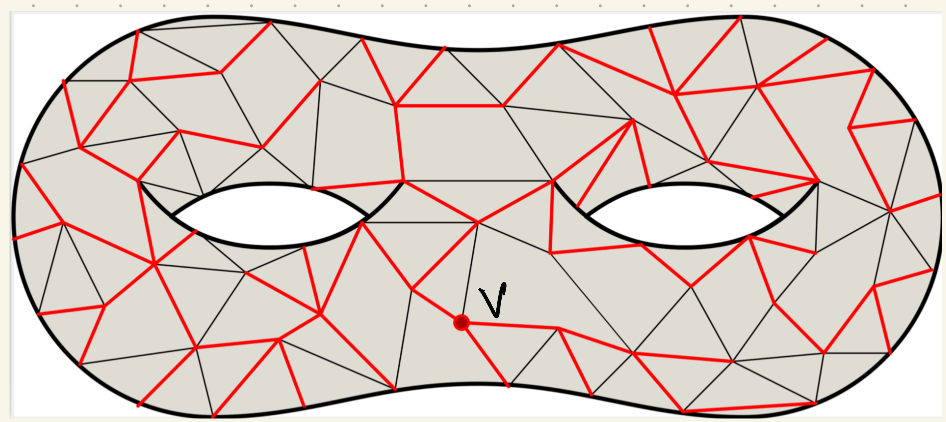
Algorithm 8 GENERATOR(K)

Input:
A 2-complex K

Output:
A set of 1-cycles containing an optimal $\mathsf{H}_1(K)$-basis

```
Let K^1 be the 1-skeleton of K with vertex set V and edge set E
𝒞 := {∅}
for all v ∈ V do
    compute a shortest path tree T_v rooted at v in K^1 = (V, E)
    for all e = (u, w) ∈ E \ T_v s.t. u, w ∈ T_v do
        Compute cycle c_e = π_{u,w} ∪ {e} where π_{u,w} is the unique path connecting u and w in T_v
        𝒞 := 𝒞 ∪ {c_e}
    end for
end for
Output 𝒞
```

If optimal basis has any cycle not in $\mathcal{C}$, can replace with on in $\mathcal{C}$.



So: If K has $O(n)$ vertices & edges

$$\Rightarrow |\mathcal{C}| =$$

Time to compute:

Then, we can sort $\mathcal{C}$, and loop through:
check if c_i is independent from optimal basis so far.

Annotations:
assignment $a: K_p \to \mathbb{Z}_2^d$, with $d = \dim(H_p(K))$, giving each simplex σ a binary vector of length d.

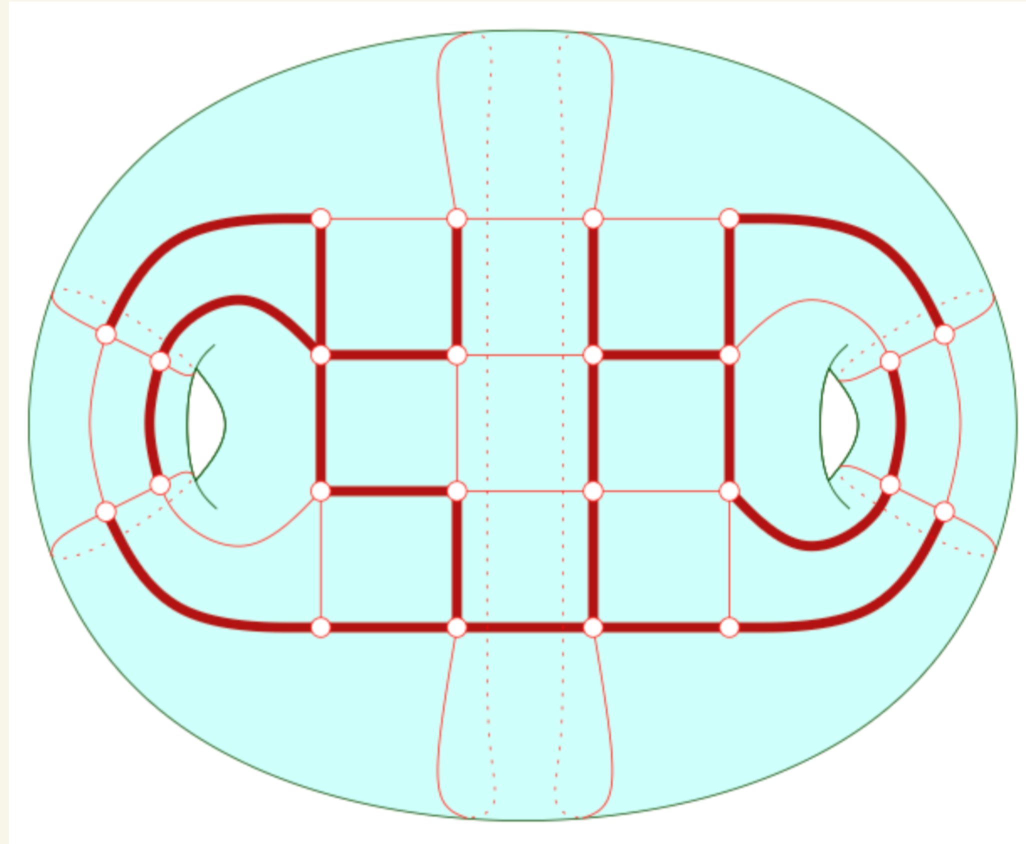
Valid if two p-cycles have same anotation iff they are homologous.

... how to create?

Algorithm 10 AnnotEdge(K)

Input:
A simplicial 2-complex K
Output:
Annotations for edges in K

Let K^1 be the 1-skeleton of K with edge set E
Compute a spanning forest T of K^1; $m = |E| - |T|$
For every edge $e \in E \cap T$, assign an m-vector $\mathsf{a}(e)$ where $\mathsf{a}(e) = 0$
Index remaining edges in $E \setminus T$ as $e_1, \ldots, e_m$
For every edge e_i, assign $\mathsf{a}(e_i)[j] = 1$ iff $j = i$
for all triangle $t \in K$ **do**
  if $\mathsf{a}(\partial t) \neq 0$ **then**
    pick any non-zero entry b_u in $\mathsf{a}(\partial t)$
    add $\mathsf{a}(\partial t)$ to every edge e s.t. $\mathsf{a}(e)[u] = 1$
    delete u-th entry from annotation of every edge
  end if
end for



Let's try:

End result:

Algorithm 7 GreedyBasis($\mathcal{C}$)

Input:

A set of p-cycles $\mathcal{C}$ in a complex

Output:

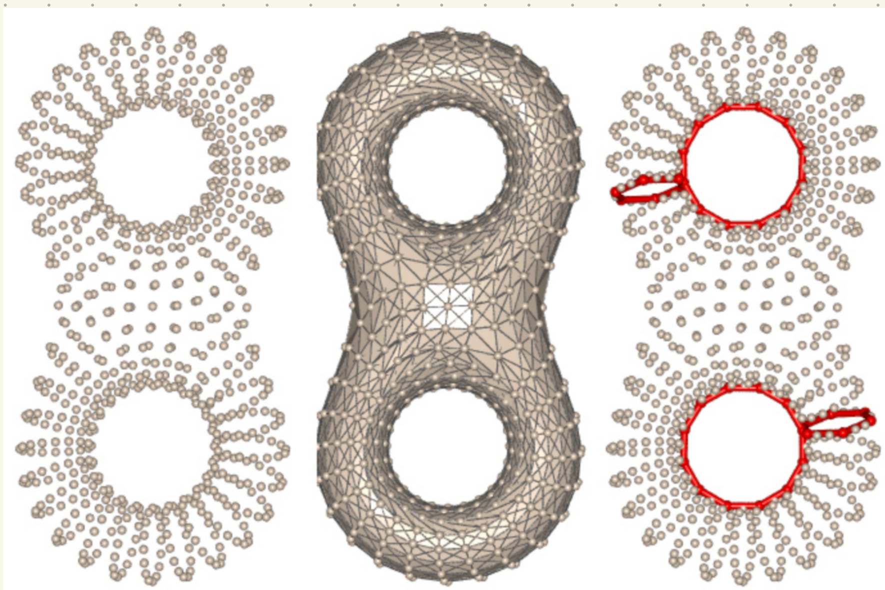
A maximal set of cycles from $\mathcal{C}$ whose classes are independent and total weight is minimum

```
Sort the cycles from C in non-decreasing order of their weights; that is, C = {c_1, ..., c_n}
    implies w(c_i) ≤ w(c_j) for i ≤ j
Let B := {c_1}
for i = 2 to n do
    if [c_i] is independent w.r.t. B then
        B := B ∪ {c_i}
    end if
end for
```

If $p=1$, use previous pieces!

Adaptation to point clouds

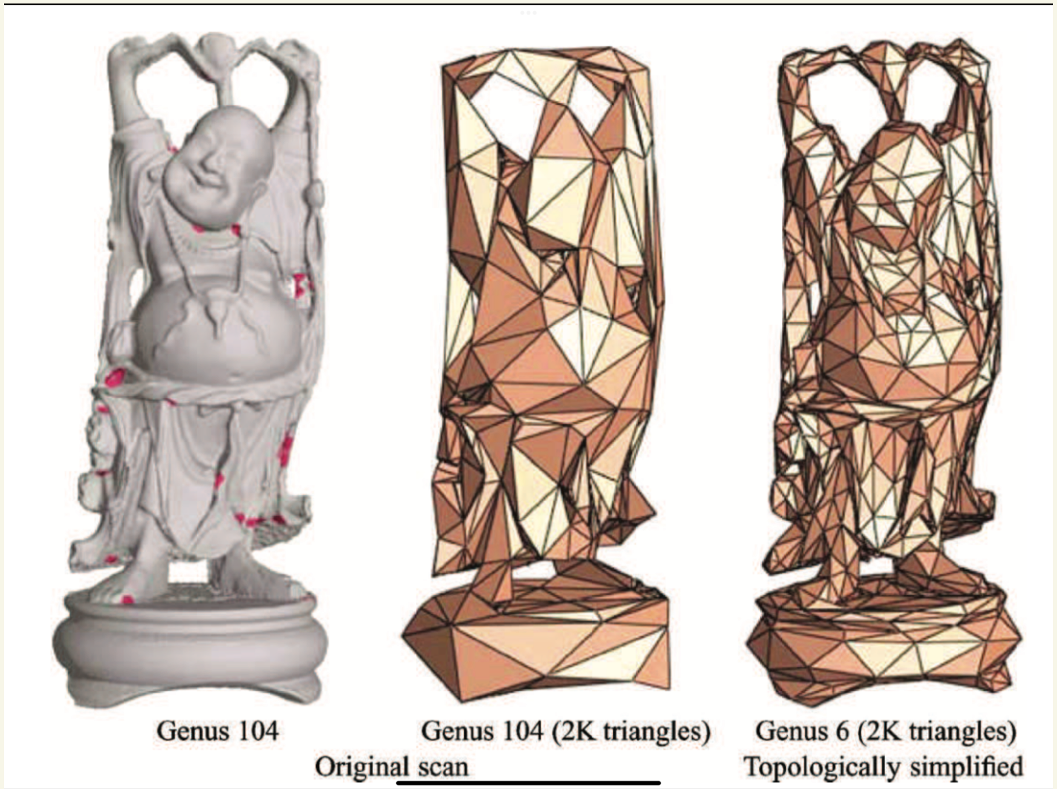
Use Rips complex $\boxed{\text{Dey-Sun-Wang 2010}}$



Not a true mesh, but result is provably ε-close to optimal one if well-sampled.

Applications

① Noise detection:



Idea:

Wood-Hoppe-Desbrun-Schroder 2004

② Surface Parameterization

Classic problem of how to project meshes into 2d:

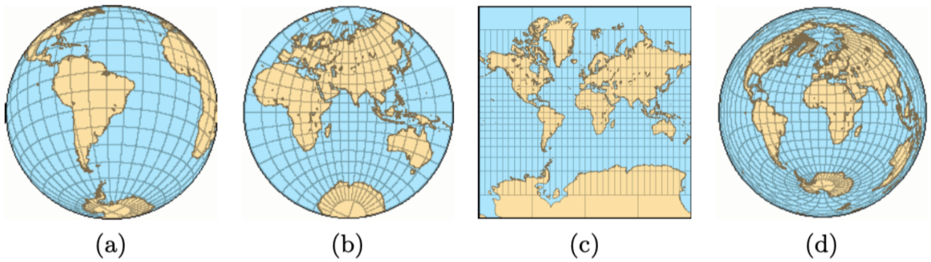
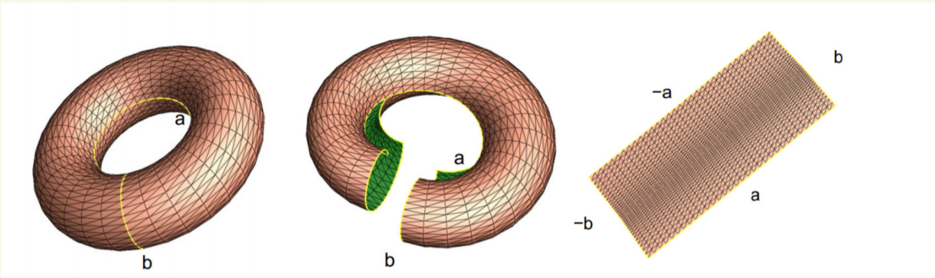


Fig. 1. Orthographic (a), stereographic (b), Mercator (c), and Lambert (d) projection of the Earth.



Many variants of this on surfaces want to cut along small curves:

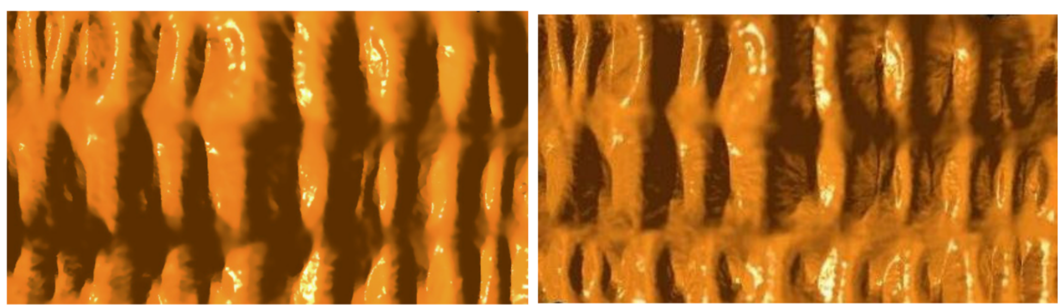


Fig. 7. Letf : The zoomed-in results from method in [9]. Right : The zoomed-in results from our method.

Shi et al 2012

(on colon data)

What about dim > 1?

Unfortunately, NP-Hard even to approximate. $\boxed{\text{Chen - Freedman 2011}}$

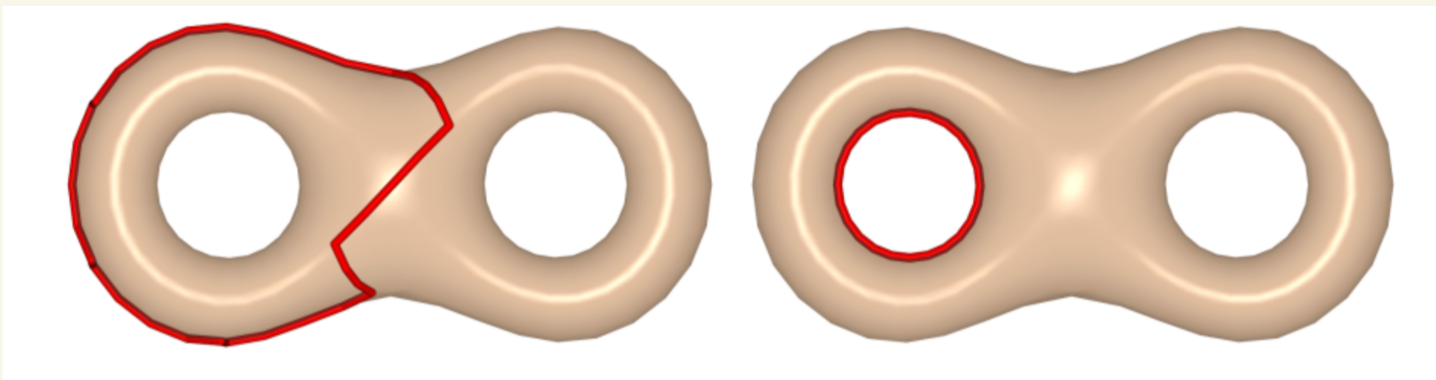
Reduction: Known NP-Hard Problem

Problem 4.2.1 (Nearest Codeword Problem)
INPUT: *an* $m \times k$ *generator matrix* A *over* $\mathbb{Z}_2$ *and a vector* $y_0 \in \mathbb{Z}_2^m \setminus \operatorname{span}(A)$
OUTPUT: *a vector* $y \in y_0 + \operatorname{span}(A)$
MINIMIZE: *the Hamming weight of* y

Key: Build a triangulation s.t. p^{th}-boundary matrix is A.

Related question: homology localization

Given a p-cycle c, find minimum weight cycle c' such that $[c'] = [c]$.



Interestingly:

- For $\mathbb{Z}_2$-homology, NP-Hard
- With $\mathbb{Z}$-coefficients, polynomial time (if no torsion)

Algorithm:

Reduce to integer programming

Given p-chain $x = \sum_{i=0}^{m-1} x_i \sigma_i$, $x_i \in \mathbb{Z}$,

let $\vec{x} \in \mathbb{Z}^m$ be $\vec{x} = (x_0, \ldots, x_i, \ldots, x_{m-1})$

Recall: $\|x\|_1 = \sum_{i=0}^{m} |x_i|$

& D_p be boundary matrix $\partial_p : C_p \to C_{p-1}$

Let W be weight matrix:

$$\begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

Why?

Take cycle x.

$$W x = \begin{bmatrix} w_{\sigma_1} & & & \\ & w_{\sigma_2} & & \\ & & \ddots & \\ & & & w_{\sigma_m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

Then $|Wx| =$

Then: ILP is

Given a p-chain c, weights W,

$$\underset{x,y}{\text{minimize}} \quad \|Wx\|_1$$

$$\text{s.t.} \quad x = c + D_{p+1} y$$

$$x \in \mathbb{Z}^m$$

$$y \in \mathbb{Z}^n$$

wher m = # of p-simplices

& n = # of (p+1)-simplices

Problem: $\underbrace{\text{Integer}}$ Linear Programming

Hard!

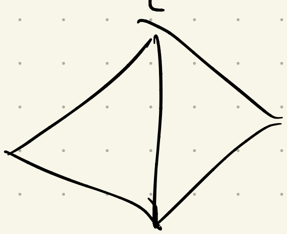
But:

If determinant of every square submatrix is $0, \pm 1$, then matrix is totally unimodular

Fact: If a matrix is totally unimodular, then the $\underbrace{\text{LP}}_{\text{polynomial time!}}$ also solves the ILP.

Claim: D_{p+1} is totally unimodular when K triangulates a $(p+1)$-dim compact orientable manifold

Why? • Each p-simplex is facet of ≤ 2 $p+1$ simplices



$\Rightarrow$ each row τ

• Known sufficiency conditions for 0-1 matrices work for D_{p+1}

$\boxed{\text{Heller-Tompkins 56}}$

Torsion

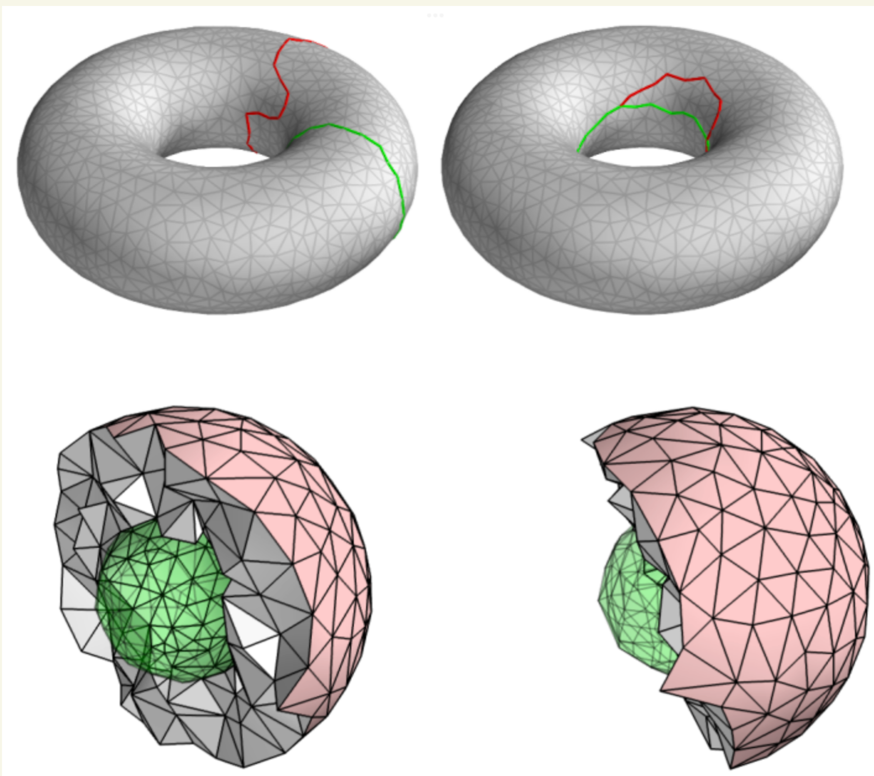
Unfortunately, not 0, 1-matrix for D_i with $i < p+1$, + fails for $\mathbb{Z}_2$ entirely.

More generally: Any group G can be written as $G = F \oplus T$

- $F \cong (\mathbb{Z} \oplus \cdots \oplus \mathbb{Z})$
- $T \cong (\mathbb{Z}/t_1 \oplus \cdots \oplus \mathbb{Z}/t_k)$

torsion subgroup

Theorem: D_{p+1} is totally unimodular $\iff$ $H_p(L, L_0)$ is torsion-free for all pure subcomplexes L_0 & L in K of dimensions p & $p+1$ respectively, where $L_0 \subset L$



Dey-Hirani-Krishnamoorthy 2011

Next time:

Optimal persistent cycles

Inside a filtration, how to get "best" cycle in a persistent homology class?

Recall: had barcodes or diagrams

but many choices of representative