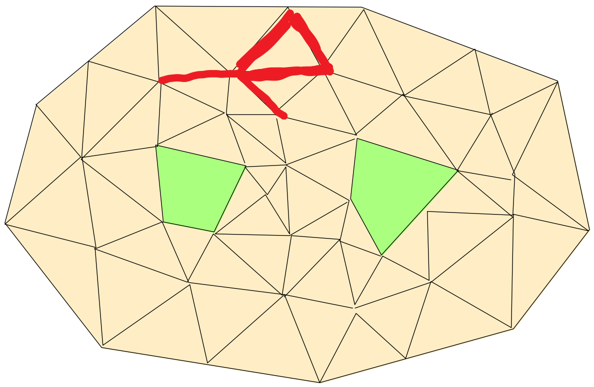
Topological Data Analysis and Neuroscience

Chapter 3: Simplicial Homology

Instructor: Alex McCleary

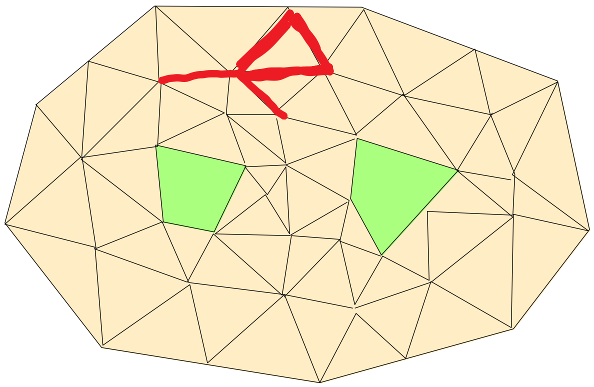
Chains

- Given a simplicial complex K, a p-chain is
  - A formal sum of p-simplices $c = \sum c_i \sigma_i$
  - Under Z_2 coefficients: a collection of p-simplices



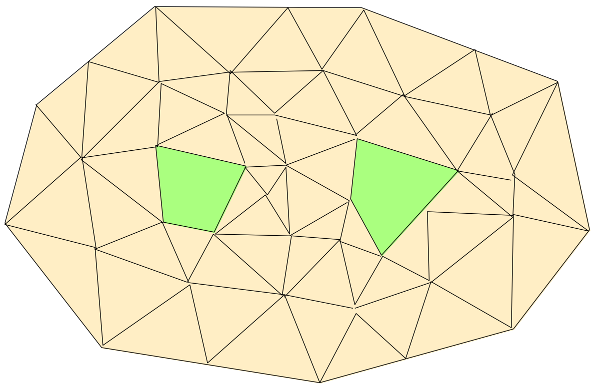
Chains

- Given a simplicial complex K, a p-chain is
  - A formal sum of p-simplices $c = \sum c_i \sigma_i$
  - Under Z_2 coefficients: a collection of p-simplices
- p-th chain group of K
  - $C_p(K)$: collection of p-chains with operation $+$
    - $c_1 = \sigma_1 + \sigma_3;\ \ c_2 = \sigma_1 + \sigma_4;\ \ \Rightarrow\ \ c_1 + c_2 = \sigma_3 + \sigma_4$
- Under Z_2 coefficients,
  - $C_p(K)$ is a vector space
  - What is its dimension (rank)?
  - What is a basis for it?



Chains and boundary operator

- p-th boundary operator $\partial_p \colon C_p \to C_{p-1}$
  - $\partial_p(\sigma) =$ sum of $(p-1)$-faces of σ
  - $\partial_p(c) = \sum c_i \partial_p(\sigma_i)$
  - Hence ∂_p is a homomorphism (map preserving + operation)



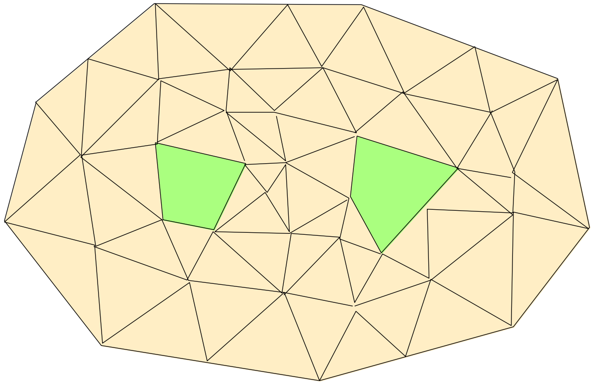
Chains and boundary operator

- p-th boundary operator $\partial_p: C_p \to C_{p-1}$
  - $\partial_p(\sigma) =$ sum of $(p-1)$-faces of σ
  - $\partial_p(c) = \sum c_i \partial_p(\sigma_i)$
  - Hence ∂_p is a homomorphism (map preserving + operation)

- Chain complex

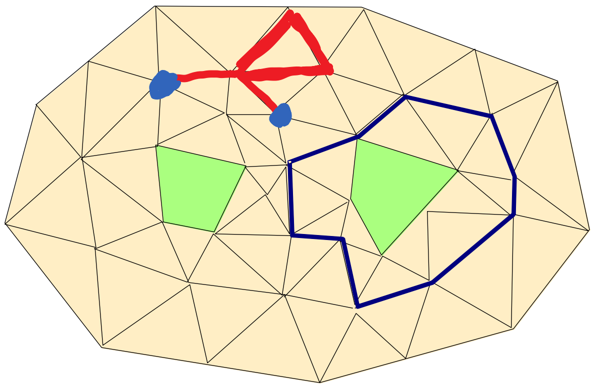
$$\cdots \xrightarrow{\partial_{p+2}} \mathrm{C}_{p+1} \xrightarrow{\partial_{p+1}} \mathrm{C}_p \xrightarrow{\partial_p} \mathrm{C}_{p-1} \xrightarrow{\partial_{p-1}} \cdots .$$

Theorem:
$\partial_p \circ \partial_{p+1} = 0$



Cycles and Boundaries

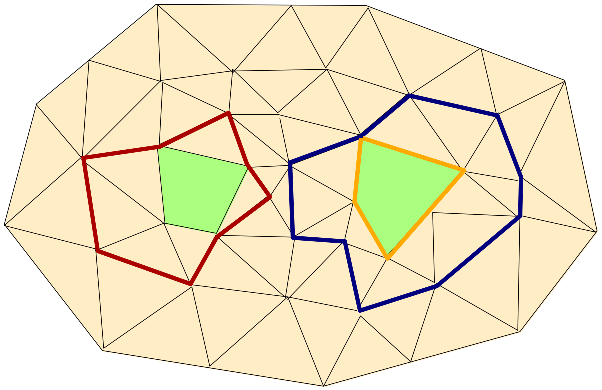
- Cycles:
  - p-cycle: a p-chain whose boundary is 0
  - p-th cycle group $Z_p(K) = \ker(\partial_p)$
  - What is the relation between Z_p and C_p ?



Cycles and Boundaries

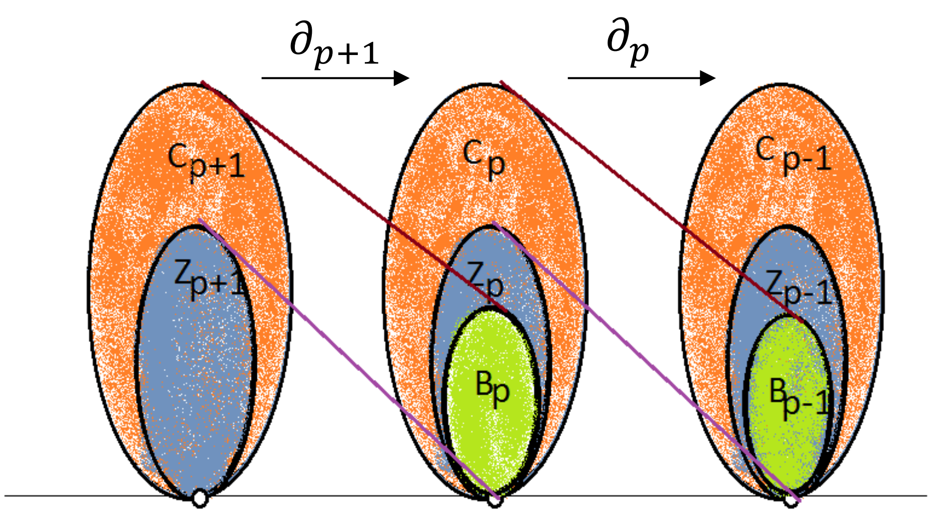
- Cycles:
  - p-cycle: a p-chain whose boundary is 0
  - p-th cycle group $Z_p(K) = \ker(\partial_p)$
- Boundary cycles:
  - p-boundary: a p-cycle which is the boundary of some $(p+1)$-chain
  - p-th boundary group $B_p(K) = \mathrm{Im}(\partial_{p+1})$
- $\partial_p \circ \partial_{p+1} = 0 \Rightarrow B_p \subseteq Z_p \subseteq C_P$

Under Z_2 coefficients, B_p, Z_p, C_p are all vector spaces.



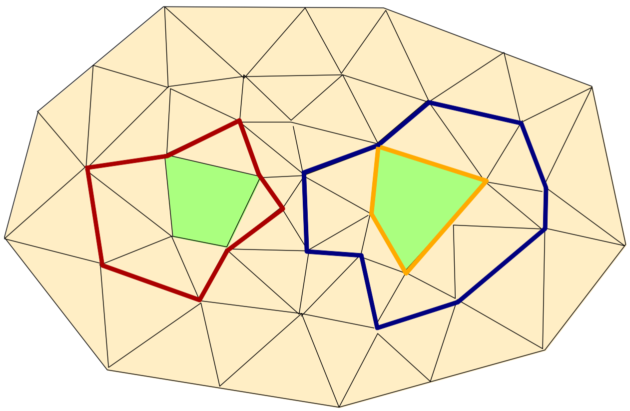
Cycles and Boundaries

- Cycles:
  - p-cycle: a p-chain whose boundary is 0
  - p-th cycle group $Z_p(K) = \ker(\partial_p)$
- Boundary cycles:
  - p-boundary: the boundary of some $(p+1)$-chain
  - p-th boundary group $B_p(K) = \mathrm{Im}(\partial_{p+1})$
- $\partial_p \circ \partial_{p+1} = 0 \Rightarrow B_p \subseteq Z_p \subseteq C_P$



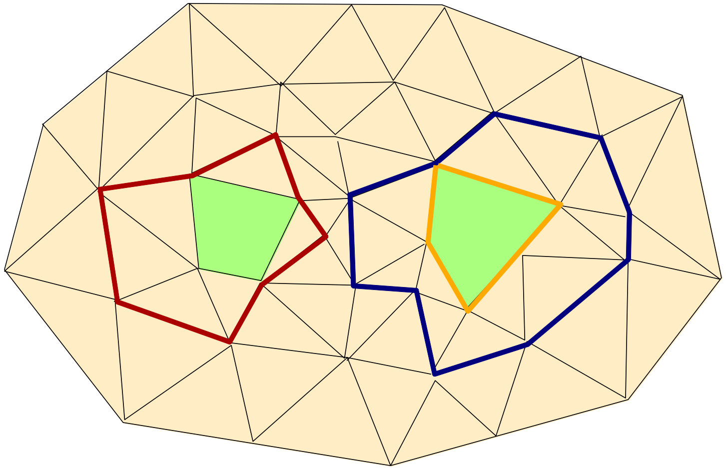
Homology groups

- p-th cycle group $Z_p(K) = \ker(\partial_p)$
- p-th boundary group $B_p(K) = \mathrm{Im}(\partial_{p+1})$
- p-th homology group
  - $H_p(K) = Z_p/B_p$
  - c_1 is homologous to c_2 if
    - $c_1 - c_2 \in B_p$, i.e, $c_1 - c_2$ is a boundary cycle
  - $h = [c] \in H_p$:
    - the family p-cycles homologous to c
    - called a homology class



Betti numbers

- Betti number: $\beta_p(K) = rank\left(H_p\right)$
- Theorem:
  - $\beta_p(K) = rank\left(Z_p\right) - rank(B_p)$
- Examples: meaning of $\beta_0, \beta_1, \beta_2$



Rank(H_0) = ?; Rank(H_1) = ?

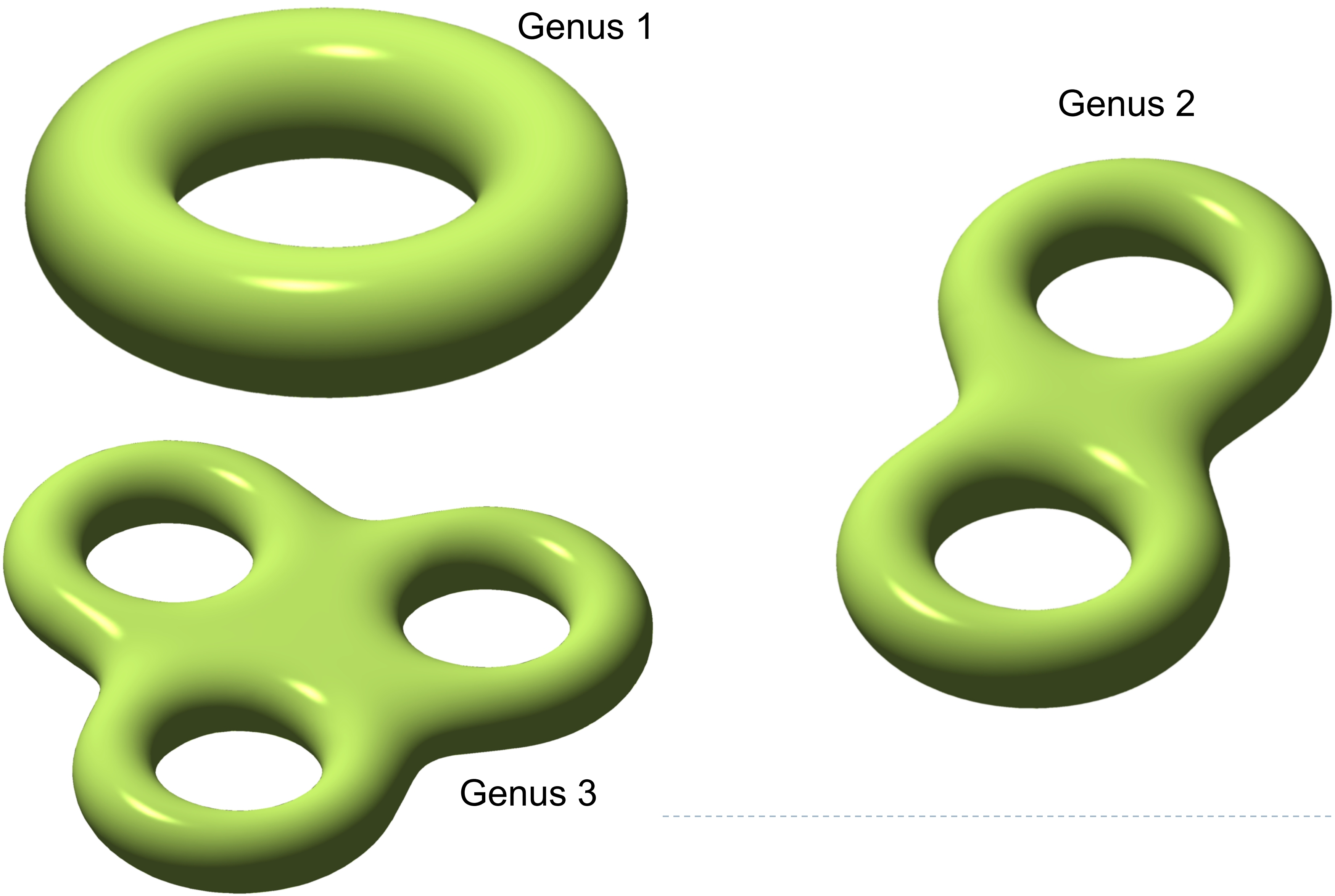
More results

- **Theorem:**
  - For a compact orientable 2-manifold with genus g, we have
    - $\beta_0 = 1, \;\; \beta_1 = 2g, \;\; \beta_2 = 1$

- **Theorem:**
  - Given two simplicial complexes K_1 and K_2 such that $|K_1| \cong |K_2|$, then $H_*(K_1) \approx H_*(K_2)$.

- Hence different triangulations of the same space have isomorphic homology groups!
- Thus homology groups are a topological invariant

Genus of a surface



Euler characteristics

- Given a topological space M
  - its Euler characteristics $\chi(M) = \sum_{p\geq 0} (-1)^p \beta_p(M)$

- Theorem (Euler-Poincaré formula)
  - Given a simplicial complexes K, let n_p denote the number of p-simplices in K. Then

$$\chi(K) \coloneqq \chi(|K|) = \sum_{p=0} (-1)^p n_p$$

- Hence Euler characteristics is also independent of the triangulation of a space, and is a topological invariant.

Examples of triangulations of 2-manifolds.

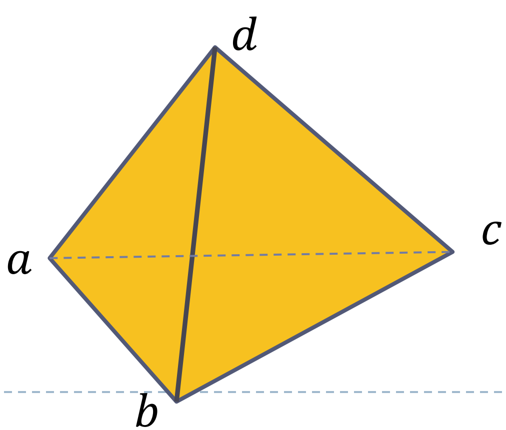
Section 2:
Matrix view and computation

Boundary Matrix

- $K^p = \left\{\alpha_1, \ldots, \alpha_{n_p}\right\}, K^{p-1} = \left\{\tau_1, \ldots, \tau_{n_{p-1}}\right\}$
  - K^p forms a basis for p-th chain group C_p
- $\mathrm{n}_{p-1} \times n_p$ boundary matrix A_p s.t.
  - $A_p[i][j] = 1$ iff $\tau_i \subseteq \sigma_j$
  - representing $\partial_p : C_p \to C_{p-1}$

Boundary Matrix

- $K^p = \left\{\alpha_1, \dots, \alpha_{n_p}\right\}, K^{p-1} = \left\{\tau_1, \dots, \tau_{n_{p-1}}\right\}$
  - K^p forms a basis for p-th chain group C_p
- $\mathrm{n}_{p-1} \times n_p$ boundary matrix A_p s.t.
  - $A_p[i][j] = 1$ iff $\tau_i \subseteq \sigma_j$
  - representing $\partial_p : C_p \to C_{p-1}$ w.r.t. basis $\left\{\alpha_1, \dots, \alpha_{n_p}\right\}$ and $\left\{\tau_1, \dots, \tau_{n_{p-1}}\right\}$



$$A_2 = \begin{array}{c} \\ ab \\ ac \\ ad \\ bc \\ bd \\ cd \end{array} \begin{array}{c} \begin{array}{cccc} abc & abd & acd & bcd \end{array} \\ \begin{pmatrix} 1 & 1 & & \\ 1 & & 1 & \\ & 1 & 1 & \\ 1 & & & 1 \\ & 1 & & 1 \\ & & 1 & 1 \end{pmatrix} \end{array}$$

Boundary Matrix

- Given a p-chain $c = \sum_{i=1}^{n_p} c_i \alpha_i$
  - Under basis K^p, vector representation of c is
  - $\vec{c} = \left[c_1, c_2, \dots, c_{n_p}\right]^T$
- Boundary $\partial_p c$ is a $(p-1)$-chain with vector representation $A_p \vec{c}$ w.r.t basis K^{p-1}

$$A_p \vec{c} = \begin{bmatrix} a_1^1 & a_1^2 & \cdots & a_1^{n_p} \\ a_2^1 & a_2^2 & \cdots & a_2^{n_p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n_{p-1}}^1 & a_{n_{p-1}}^2 & \cdots & a_{n_{p-1}}^{n_p} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n_p} \end{bmatrix}$$

Example.

Boundary Matrix

- Let n_p, z_p, b_p denote the rank of C_p, Z_p, and B_p
- $\beta_p = rank(H_p)$

Boundary Matrix

- Let n_p, z_p, b_p denote the rank of C_p, Z_p, and B_p
- $\beta_p = rank(H_p)$
- Claim: $(i) \quad n_p = z_p + b_{p-1};$

  $(ii) \; \beta_p = z_p - b_p$

Boundary Matrix

- Let n_p, z_p, b_p denote the rank of C_p, Z_p, and B_p
- $\beta_p = rank(H_p)$
- Claim: $(i) \quad n_p = z_p + b_{p-1}$;

  $(ii) \ \ \beta_p = z_p - b_p$

- Consider A_p
  - Each columns of A_p corresponds to a boundary cycle
  - Rank of A_p gives $b_p = rank(B_p)$
    - Why?

Boundary Matrix

- Let n_p, z_p, b_p denote the rank of C_p, Z_p, and B_p
- $\beta_p = rank(H_p)$
- Claim: $(i) \quad n_p = z_p + b_{p-1}$;

  $(ii) \ \beta_p = z_p - b_p$

- Consider A_p
  - Each columns of A_p corresponds to a boundary cycle
  - Rank of A_p gives $b_p = rank(B_p)$
- Note, this gives an $O(n^3)$ time algorithm for computing all β_p's via Gaussian elimination
  - Can be improved to matrix multiplication time

Right-reduction algorithm

- Starting with boundary matrix $M = A_p$
  - For the i-th column corresponding to p-simplex σ_i,
    - associate a p-chain Γ_i initialized to σ_i
  - AddColumn(j, i)
    - $Col_M[i] = Col_M[i] + Col_M[j]$; $\Gamma_i = \Gamma_i + \Gamma_j$

Algorithm 1 Right-Reduction(M)

```
for i = 2 to n_p do
  while ∃j < i s.t. lowId[j] = lowId[i] do
    AddColumn(j, i);
  end while
end for
Return(M)
```

Properties

- Lemma:
  - Each reduction (column addition) step maintains the following invariance: After k-th stages, $M^{(k)}$ has the same rank as A_p, and $\partial_p \Gamma_j^{(k)} = col_M[j]$ for any j.

- Lemma:
  - At the end of the reduction algorithm, each non-zero column has a unique low-ID.

- Reduced form:
  - A matrix M is in reduced form is all non-zero columns are linearly independent.
  - It is in Smith-Normal form if it has the following structure:

$$S_p = \begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & \ddots & & & 0 & \\ & & & 1 & & & \\ & & & & 0 & & \\ & & 0 & & & \ddots & \\ & & & & & & 0 \end{bmatrix}$$

- Lemma:
  - A matrix is in reduced form if each non-zero column has a unique low-ID.

Properties

- Theorem:
  - Procedure Left-Reduction(M) terminates in $O(n_p^2 n_{p-1})$ time
  - The output matrix M is in reduced form
  - The set of non-zero columns in M form a basis for B_{p-1}
  - The set $\{\Gamma_i \mid col_M[i] = 0\}$ form a basis for Z_p

Examples.

This is not the only reduction algorithm!! Any elimination via row/column additions to convert a matrix into a reduced form works!