Computational topology: Lecture 5

Alberto Paoluzzi

March 15, 2019

1. Chain Complexes
2. Topology computing with chains
3. Boundary and coboundary

Chain Complexes

Chain groups

Chains are defined by attaching coefficients to cells.

Since one wishes to add chains, one has to pick coefficients from a set endowed with the structure of a commutative group, or stronger.

Let $(G, +, 0)$ be a nontrivial commutative group, whose identity element is denoted 0.

A p-chain of X with coefficients in G is a mapping $c_p : X \to G$ such that, for each $\sigma \in X_p$, reversing a cell orientation changes the sign of the chain value:

$$c_p(-\sigma) = -c_p(\sigma).$$

Chain groups

Chain addition is defined by addition of chain values: if c_p^1, c_p^2 are p-chains, then $(c_p^1 + c_p^2)(\sigma) = c_p^1(\sigma) + c_p^2(\sigma)$, for each $\sigma \in X_p$.

The resulting group is denoted $C_p(X; G)$. When clear from the context, the group G is often left implied, writing $C_p(X)$.

Let σ be an oriented cell in X and $g \in G$. The {elementary chain} whose value is g on σ, $-g$ on $-\sigma$ and 0 on any other cell in X is denoted $g\sigma$.

Each chain can be written in a unique way as a sum of elementary chains.

Chain groups

Chains are often thought of as attaching orientation and/or multiplicity to cells

If coefficients are taken from the group $G = (\{-1, 0, 1\}, +, 0)$, then cells can only be discarded or selected, possibly inverting their orientation

A p-cycle is a closed p-chain, i.e., a p-chain without boundary.

Examples

- a 2D polygon boundary
- a 2-manifold-without-boundary in 3D

Chain orientation

It is useful to select a conventional orientation to orient cells automatically.

- 0-cells are considered all positive.
- Closed p-cells can be given a coherent (internal) orientation according with the orientation of the first $(p-1)$-cell in their canonical representation (sorted on indices) of their $(p-1)$-cycles.
- Finally, a d-cell may be oriented as the sign of its oriented volume.

Chain spaces

To allow not only for chain addition, but also for linear combination of chains, coefficients should be taken from a set endowed with the structure of a field, such as $(\mathbb{F}, +, \times, 0, 1)$,

where 0 and $1 \neq 0$ denote, respectively, the additive and multiplicative identities.

Unit chains are elementary chains whose value is $u = 1\sigma$ for some cell σ.

Each chain can be written in a unique way as a linear combination of unit chains $u \in U$, if the outer cell is not taken into account.

Chain spaces

Canonical representation and standard basis

Hence, the space of p-chains C_p is endowed with a standard (or natural) basis, comprised of all the independent unit p-chains.

The standard basis may be the one associated with the canonical representation . . .

In particular, $\#U_d = \#\Lambda_d - 1$.

Often, with some abuse of notation, one does not distinguish between a p-cell and the corresponding unit p-chain.

Characteristic matrices

Given a set $S = \{s_j\}$, the characteristic function $\chi_A : S \to \{0, 1\}$ takes value 1 for all elements of $A \subseteq S$ and 0 at all elements of S not in A.

We call characteristic matrix M of a collection of subsets $A_i \subseteq S$ $(i = 1, \ldots, n)$ the binary matrix $M = (m_{ij})$, with $m_{ij} = \chi_{A_i}(s_j)$.

A matrix M_p, whose rows are indexed by unit p-chains and columns are indexed by unit 0-chains, provides a useful representation of a basis for the linear space C_p.

Permuting (reindexing) either rows or columns provides a different basis.

While chains are mostly presented as formal sums of cells, in the actual implementation their signed coordinate vectors are used as sparse arrays, and in particular as CSC (Compressed Sparse Column) maps : $\mathbb{N} \to \{-1, 0, 1\}$.

Characteristic matrices

Implementation

Implementation

How ?

help?> Lar.characteristicMatrix

WHY Characteristic matrices are useful

Consider the incidence queries that arise in a cellular decomposition $\Lambda(X)$ of a 2D space, such as 2D triangulation or the boundary of a manifold 3-solid.

There are 9 incidence relations between pairs of cells in such a V, E, F decomposition, and traditional graph-based representations are chosen by optimizing the trade-off between space requirements and efficiency of queries.

With LAR, all such queries are sparse matrix–vector multiplications without any additional space requirements.

The chain operators corresponding to the incidence relations are

$$\begin{array}{lll} \mathcal{V}\mathcal{V} : C_0 \to C_0, & \mathcal{E}\mathcal{V} : C_0 \to C_1, & \mathcal{F}\mathcal{V} : C_0 \to C_2; \\ \mathcal{V}\mathcal{E} : C_1 \to C_0, & \mathcal{E}\mathcal{E} : C_1 \to C_1, & \mathcal{F}\mathcal{E} : C_1 \to C_2; \\ \mathcal{V}\mathcal{F} : C_2 \to C_0, & \mathcal{E}\mathcal{F} : C_2 \to C_1, & \mathcal{F}\mathcal{F} : C_2 \to C_2. \end{array}$$

Characteristic matrices: Examples

From incidence/adjacency operators to their matrices

The corresponding CSR matrices are readily computed:

$$\mathcal{V}\mathcal{V} = \mathcal{V}\mathcal{E} \circ \mathcal{E}\mathcal{V} = \mathcal{E}\mathcal{V}^\top \circ \mathcal{E}\mathcal{V} \Rightarrow [\mathcal{V}\mathcal{V}] = M_1^t M_1$$

$$\mathcal{V}\mathcal{E} = \mathcal{E}\mathcal{V}^\top \Rightarrow [\mathcal{V}\mathcal{E}] = M_1^t$$

$$\mathcal{V}\mathcal{F} = \mathcal{F}\mathcal{V}^\top \Rightarrow [\mathcal{V}\mathcal{F}] = M_2^t$$

$$\mathcal{E}\mathcal{V} \quad [\mathcal{E}\mathcal{V}] = M_1$$

$$\mathcal{E}\mathcal{E} = \mathcal{E}\mathcal{V} \circ \mathcal{V}\mathcal{E} = \mathcal{E}\mathcal{V} \circ \mathcal{E}\mathcal{V}^\top \Rightarrow [\mathcal{E}\mathcal{E}] = M_1 M_1^t$$

$$\mathcal{E}\mathcal{F} = \mathcal{E}\mathcal{V} \circ \mathcal{V}\mathcal{F} = \mathcal{E}\mathcal{V} \circ \mathcal{F}\mathcal{V}^\top \Rightarrow [\mathcal{E}\mathcal{F}] = M_1 M_2^t$$

$$\mathcal{F}\mathcal{V} \quad [\mathcal{F}\mathcal{V}] = M_2$$

$$\mathcal{F}\mathcal{E} = \mathcal{F}\mathcal{V} \circ \mathcal{V}\mathcal{E} = \mathcal{F}\mathcal{V} \circ \mathcal{E}\mathcal{V}^\top \Rightarrow [\mathcal{F}\mathcal{E}] = M_2 M_1^t$$

$$\mathcal{F}\mathcal{F} = \mathcal{F}\mathcal{V} \circ \mathcal{V}\mathcal{F} = \mathcal{F}\mathcal{V} \circ \mathcal{F}\mathcal{V}^\top \Rightarrow [\mathcal{F}\mathcal{F}] = M_2 M_2^t.$$

Characteristic matrices: Examples

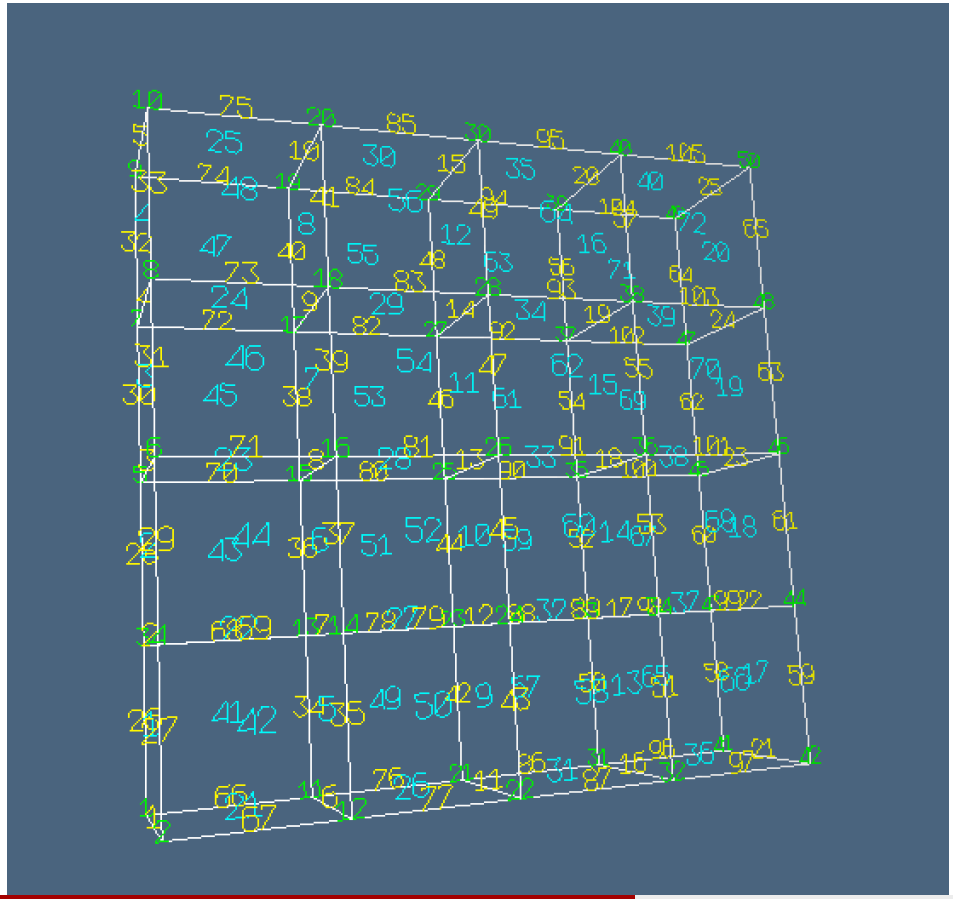
```
function characteristicMatrix( FV::Lar.Cells )::Lar.ChainOp
    I,J,V = Int64[],Int64[],Int8[]
    for f=1:length(FV)
        for k in FV[f]
            push!(I,f)
            push!(J,k)
            push!(V,1)
        end
    end
    M_2 = sparse(I,J,V)
    return M_2 # optional
end
```

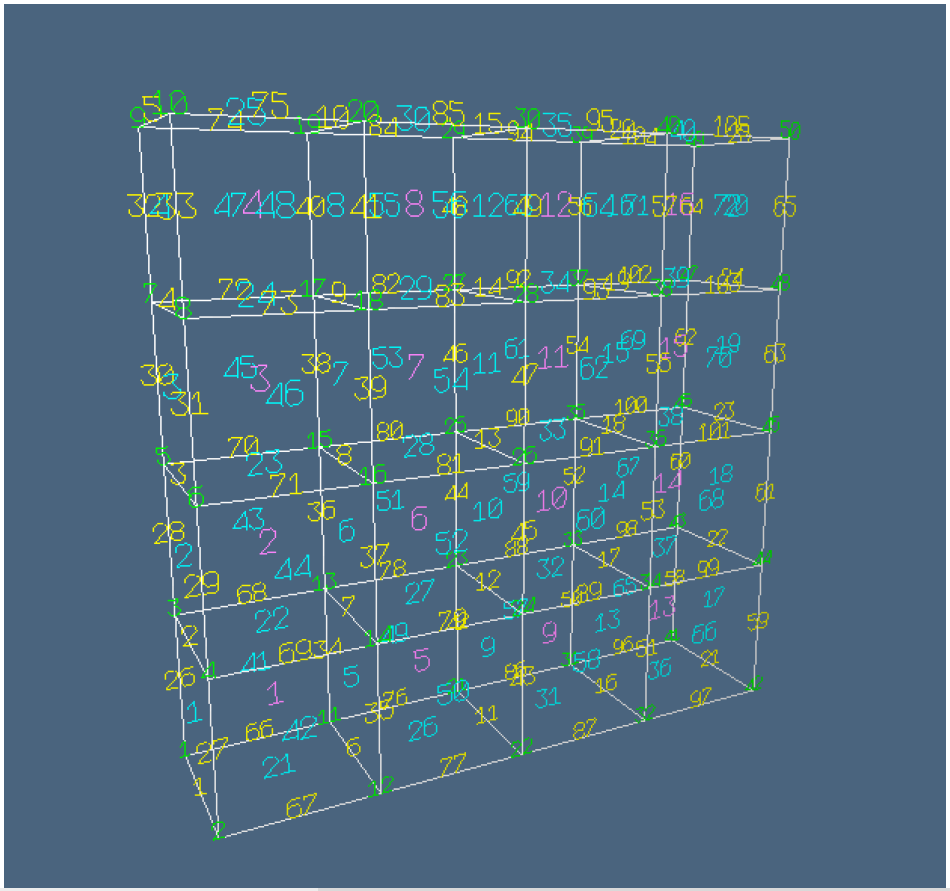
```
julia> V,(VV,EV,FV,CV) = Lar.cuboid([.5,.5,.5], true) # single 3-cube
julia> Matrix(Lar.characteristicMatrix(FV))
julia> Matrix(Lar.characteristicMatrix(FV))

julia> V,(VV,EV,FV,CV) = Lar.cuboidGrid([3,3], true); # grid of unit 2-cube
```

Example: numbered cuboidGrid

```
help?> Plasm.numbering
```

```
Plasm.view(Plasm.numbering(1.)((V,[VV,EV,FV])))
Plasm.view(Plasm.numbering(1.)((V,[VV,EV,FV,CV])))
```





Cochain spaces

Cochains are dual to chains: p-cochains map linearly p-chains to the underlying field $\mathbb{F}$.

Unit p-cochains, that yield 1 when evaluated on one unit p-chain and 0 when evaluated on all the others, form the standard basis of the space of p-cochains C^p.

The linear spaces C_p and C^p, being isomorphic, can be identified with each other in infinitely many ways.

Different legitimate identifications, while affecting the metric properties of the chain-cochain complex, do not change the topology of finite complexes. [1]

[1] Since we shall use only the topological properties of finite chain-cochain complexes defined by piecewise linear cell complexes in Euclidean space, we feel free to chose the simplest possible identification, consisting in identifying each element of the standard basis of C_p with the corresponding element of the standard basis of C^p. In this paper, we take for granted that chains and cochains are identified in this trivial way.

Topology computing with chains

Example A.2 (Chains)

Unoriented chains take coefficients from $\mathbb{Z}/2\mathbb{Z} = \mathbb{Z}_2 = \{0, 1\}$.
$c \in C_0$ is given by $c = 1\nu_1 + 1\nu_2 + 1\nu_3 + 1\nu_5$. Hence, the coefficients associated to all other cells are zero.
$[1, 1, 1, 0, 1, 0]^t$ is the coordinate vector of c with respect to the (ordered) basis $(u_1, u_2, \ldots, u_6) = (1\nu_1, 1\nu_2, \ldots, 1\nu_6)$.
Analogously for the 1-chain $d \in C_1$ and the 2-chain $e \in C_2$, written as $d = \eta_2 + \eta_3 + \eta_5$ and $e = \gamma_1 + \gamma_3$, with coordinate vectors $[0, 1, 1, 0, 1, 0, 0, 0]^t$ and $[1, 0, 1]^t$, respectively.

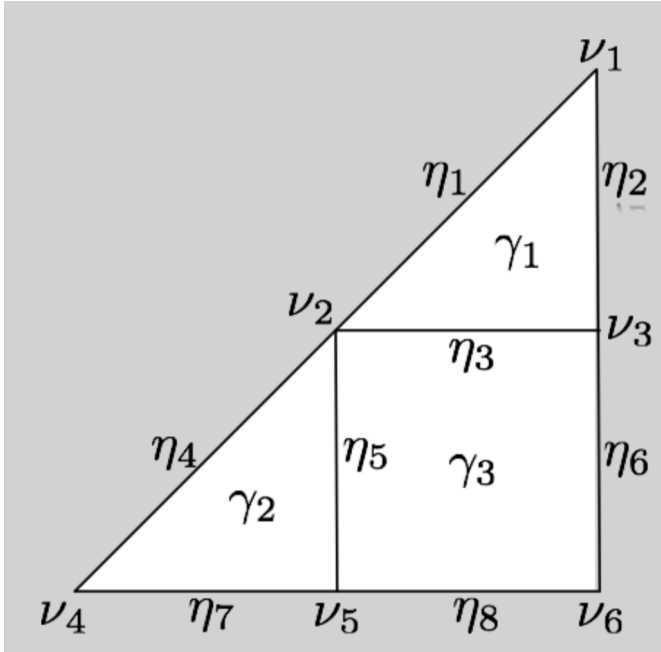


Figure 5: Cellular 2-complex

Example A.3 (Orientation).

Oriented version of the cellular complex
$\Lambda = \Lambda_0 \cup \Lambda_1 \cup \Lambda_2$
1-cells are oriented from vertex with lesser index to vertex with greater index, and all 2-cells are counterclockwise oriented
The orientation of each cell may be fixed arbitrarily, since it can always be reversed by the associated coefficient, now taken from $\{-1, 0, +1\}$
The oriented 1-chain w first vertex ν_1 and last vertex ν_5 is given as $d' = \eta_2 - \eta_3 + \eta_5$, with coordinate vector $[0, 1, -1, 0, 1, 0, 0, 0]^t$

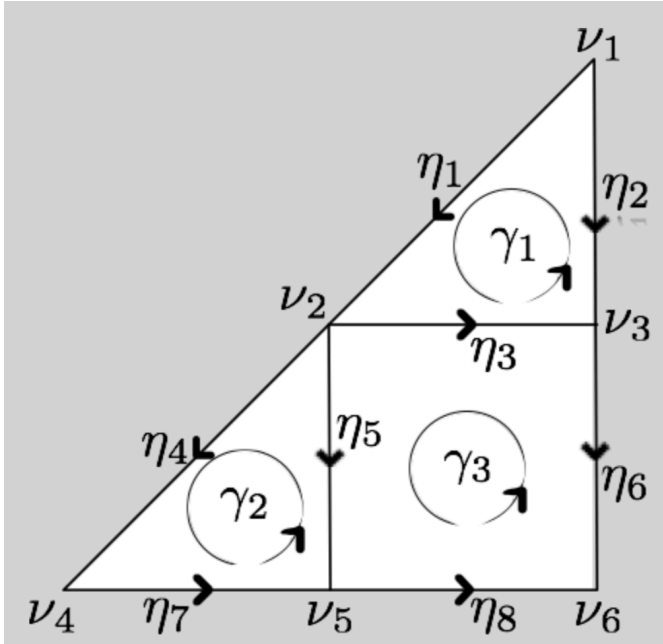


Figure 6: Oriented 2-complex

Example A.4 (Dual cochains)

0-cells are given arbitrary numbers, e.g., the values of an arbitrary scalar field;
the values for 1-cells and 2-cells are computed from these using the coboundary relations
$\boxed{\delta_0 = \partial_1^\top}$ and $\boxed{\delta_1 = \partial_2^\top}$
They are discrete gradients and curl values associated to 1-cells and 2-cells, respectively.
In discrete geometric calculus, cochains are functions from chains to reals.
Colored numbers on 1- and 2-cells are exactly the evaluation $\phi^k(u_k) = \langle \phi^k, u_k \rangle$ of the dual elementary cochain on each elementary chain

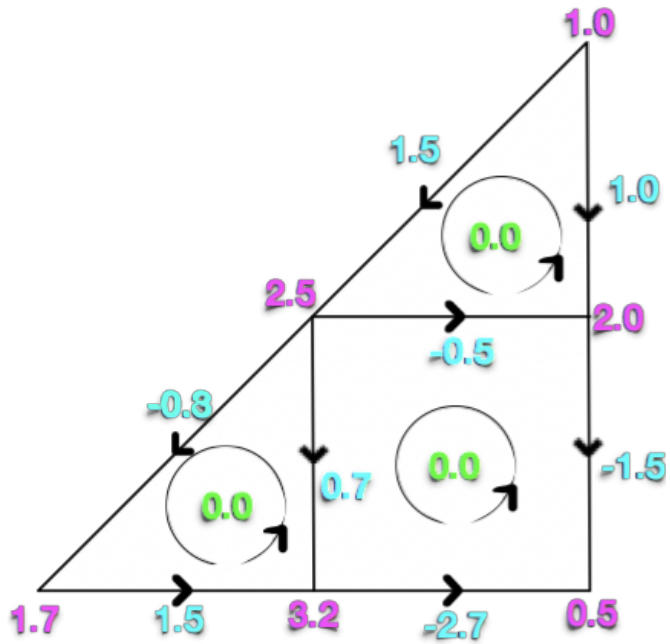


Figure 7: Elementary cochains

Definition (Boundary)

Boundary operators are maps $C_p \to C_{p-1}$, with $1 \leq p \leq d$.

Hence for a 2-complex we have two operators, denoted as

$$\partial_2 : C_2 \to C_1 \quad \text{and} \quad \partial_1 : C_1 \to C_0$$

As linear maps between linear spaces, may be represented by matrices of coefficients $[\partial_2]$ and $[\partial_1]$ from the underlying field $\mathbb{F}$.

For the unsigned and the signed case (see previous slides) we have:

$$[\partial_2] = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \text{ and } \quad [\partial_2'] = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{1}$$

Example (Boundary)

Analogously, for the unsigned ∂_1 and the signed ∂_1' operators we have:

$$[\partial_1] = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}, \quad [\partial_1'] = \begin{pmatrix} -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Example A.5 (Boundary)

As a check, let us compute:

1. the 0-boundary of the coordinate representations of the unsigned 1-chain $[d] = [0, 1, 1, 0, 1, 0, 0, 0]^t$ and
2. the signed 1-chain $[d'] = [0, 1, -1, 0, 1, 0, 0, 0]^t$

$$\partial_1 d = [\partial_1][d] \ mod\ 2 = [1, 0, 0, 0, 1, 0]^t = \nu_1 + \nu_5 \in C_0,$$

where the matrix product is computed *mod* 2, and

$$\partial'_1 d' = [\partial'_1][d'] = [-1, 0, 0, 0, 1, 0]^t = \nu_5 - \nu_1 \in C'_0.$$

Example A.5 (Boundary)

```
julia> B_1 = [
        1  1  0  0  0  0  0  0;
        1  0  1  1  1  0  0  0;
        0  1  1  0  0  1  0  0;
        0  0  0  1  0  0  1  0;
        0  0  0  0  1  0  1  1;
        0  0  0  0  0  1  0  1]
6×8 Array{Int64,2}:
...

julia> d = [0,1,1,0,1,0,0,0];

julia> B_1 * d
6-element Array{Int64,1}:
1 2 2 0 1 0

julia> B_1 * d .% 2
6-element Array{Int64,1}:
1 0 0 0 1 0
```

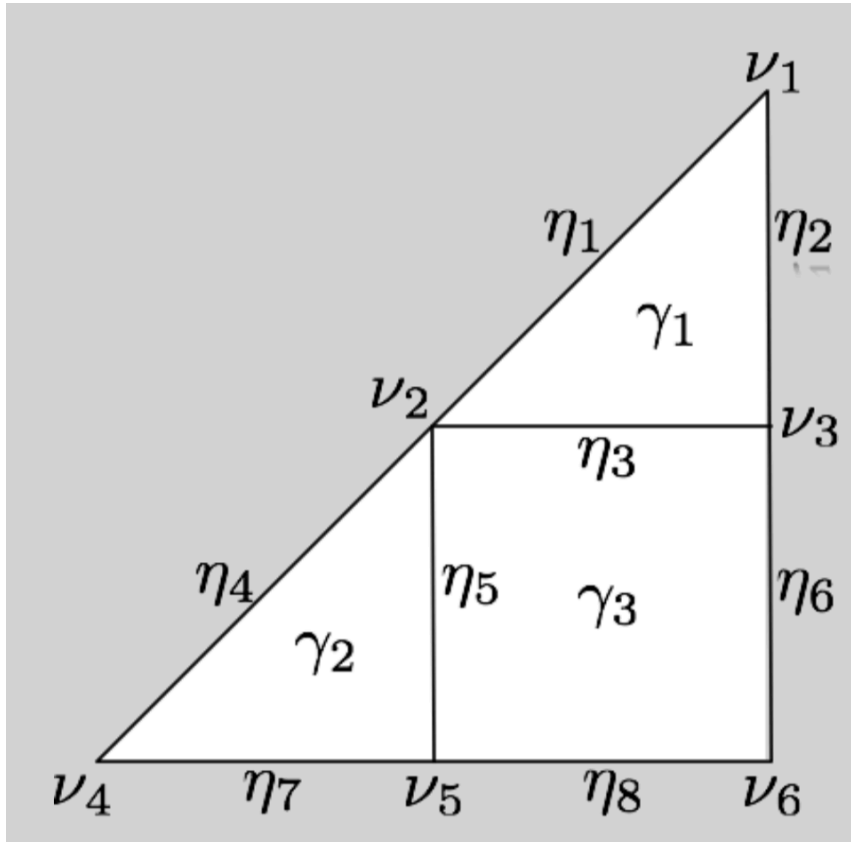


Figure 8: Cellular 2-complex

Example A.5 (Boundary)

```
julia> B_1 = [
       -1 -1  0  0  0  0  0  0;
        1  0 -1 -1 -1  0  0  0;
        0  1  1  0  0 -1  0  0;
        0  0  0  1  0  0 -1  0;
        0  0  0  0  1  0  1 -1;
        0  0  0  0  0  1  0  1]
6×8 Array{Int64,2}:
...

julia> d = [0,1,-1,0,1,0,0,0]
8-element Array{Int64,1}:
0 1 -1 0 1 0 0 0

julia> B_1 * d
6-element Array{Int64,1}:
-1 0 0 0 1 0
```

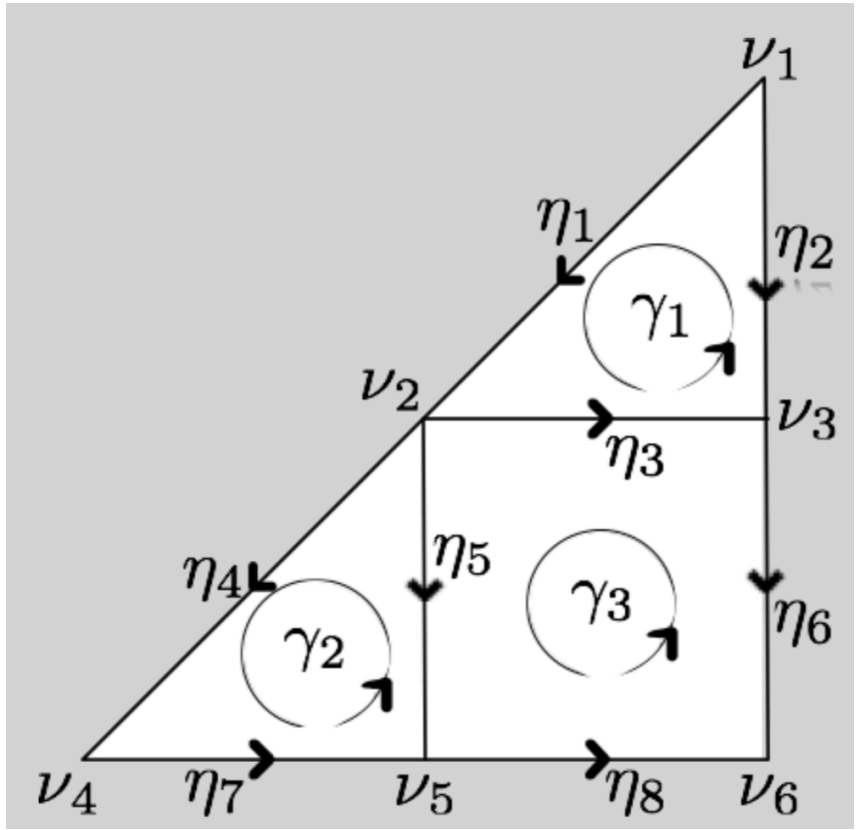


Figure 9: Cellular 2-complex

Example A.6 (Cell with a hole).

```
V = [[0.,0.],[3.,3.],[1.,2.],[2.,1.],[3.,0.],[1.,1.],[0.,3.],[2.,2.]]
FV = [[1,2,3,4,5,6,7,8],[3,4,6,8]]
EV = [[1,5],[1,7],[2,5],[2,7],[3,6],[3,8],[4,6],[4,8]]
```

```
M_1 = [
1  0  0  0  1  0  0  0;
1  0  0  0  0  0  1  0;
0  1  0  0  1  0  0  0;
0  1  0  0  0  0  1  0;
0  0  1  0  0  1  0  0;
0  0  1  0  0  0  0  1;
0  0  0  1  0  1  0  0;
0  0  0  1  0  0  0  1]
```

```
M_2 = [
1  1  1  1  1  1  1  1;
0  0  1  1  0  1  0  1]
```

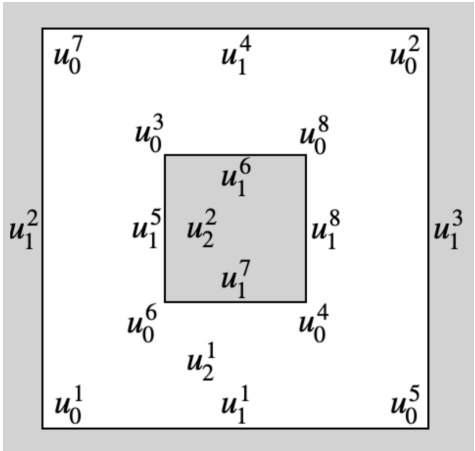


Figure 10: Cellular 2-complex

```
convert(
Array{Int64,2},(M_1 * M_2')
8×2 Array{Int64,2}:
 1  0
 1  0
 1  0
 1  0
 1  1
 1  1
 1  1
 1  1
```

```
julia> (B_2 * [1,1]) .% 2
8-element Array{Int64,1}:
1 1 1 1 0 0 0 0
```

Boundary and coboundary

Coboundary operator

The coboundary operator $\delta^p : C^p \to C^{p+1}$ acts on p-cochains as the dual of the boundary operator ∂_{p+1} on $(p+1)$-chains. For all $\phi^p \in C^p$ and $c_{p+1} \in C_{p+1}$:

$$\langle \delta^p \phi^p, c_{p+1} \rangle = \langle \phi^p, \partial_{p+1} c_{p+1} \rangle.$$

Since chain-cochain duality means integration, this defining property is the combinatorial archetype of Stokes' theorem.

See also, $(\delta^1 \circ \delta^0)(\gamma_1) = 0$.

This property, i.e., $\delta \circ \delta = 0$, is the discrete archetype of the fact that the curl of gradient is zero.

Note that a scalar field, in the discrete version, becomes a real valued 0-cochain to be valued on 0-chains, i.e., on 0-cells.

Since we use dual bases, matrices representing dual operators are the transpose of each other: for all $p = 0, \ldots, d-1$:

$$[\delta^p]^t = [\partial_{p+1}]$$