# Decomposition of Games à la Candogan

Davide Legacci

September 13, 2023

## Contents



## 1 Combinatorial Hodge Theory

The decomposition proposed by [Can+11] leverages the combinatorial version of Hodge theorem. This was introduced in [Eck44]; see [Fri96] for a concise overview, [Jia+11] for an operational introduction, and [Mun84] for a through treatment.

The idea is the following.

### 1.1 Chain complex

A normal form game can be represented as a graph, called *response graph* - see Sec. (2) - and a graph is the space on which the Hodge decomposition takes place. To make this precise we need to develop some notions of simplicial homology.

**Chain groups** A graph is an example of a *simplicial complex* $K$ - a collection of oriented $k$-dimensional faces (points, segments, triangles, tetrahedrons, ...). Given a simplicial complex $K$ one can build a family of vector spaces $C_k$, with $k$ ranging from $0$ to the dimension of the biggest face appearing in $K$. Each $C_k$ is (formally) the space of $\mathbb{R}$-linear combinations of oriented $k$-dimensional faces, i.e. the vector space spanned by the $k$-dimensional faces of the complex. These vector spaces are called *chain groups*.

An oriented $k$-dimensional face in $K$ contains $k+1$ vertices $v_0, \ldots, v_k$ and is denoted by $[v_0 \ldots v_k]$. A basis $B_k$ for $C_k$ is given by the oriented $k$-dimensional faces in the complex $K$, so the dimension of $C_k$ is the number $n_k$ of such $k$-dimensional faces. A generic vector in $C_k$ is called *k-chain* and is then a formal[1] linear combination

$$u_k = \sum_{[e] \in B_k} \lambda_k^e \, [e] \in C_k \tag{1}$$

where $\lambda_k^e$ are $n_k$ real numbers. Oriented $k$-dimensional faces, that is basis elements in $B_k$, are sometimes called *elementary k-chains*.

In the present application we need to work with $C_0, C_1$, and $C_2$:

- $C_0$ is the space of linear combinations of elementary 0-chains, i.e. vertices. Equivalently, it is the space of assignments of a number to each vertex in the complex.
- $C_1$ is the space of linear combinations of elementary 1-chains, i.e. edges. Equivalently, it is the space of assignments of a number to each edge in the complex.

  *Remark* 1.1. In the language of [Can+11], an element in $C_1$, that is a 1-chain, is called *flow*.
- $C_2$ is the space of linear combinations of elementary 2-chains (equivalently, the assignment of a number to each 2-face in the complex). Think of an elementary 2-chain as an oriented "filled" 3-vertex clique[2].

**Example 1.2.**

$$K = \begin{tikzcd} & v_2 \arrow[dr] & \\ v_0 \arrow[ur] \arrow[r] \arrow[ur, phantom, shift right=3, "\circlearrowleft"] & v_1 \arrow[u] \arrow[r] \arrow[ur, phantom, shift left=1, "\circlearrowleft"] & v_3 \end{tikzcd}$$

Here $K$ is a simplicial complex with 4 vertices (0-faces), 5 oriented edges (1-faces) and 2 oriented 2-faces. Dropping the $v$ from the symbol of each vertex for notational simplicity, the basis of $C_0$, $C_1$ and $C_2$ are respectively

$$B_0 = \{[0], [1], [2], [3]\}$$

$$B_1 = \{[01], [02], [12], [13], [23]\}$$

$$B_2 = \{[012], [123]\}$$

Generic vectors (or chains) in these spaces are

$$u_0 = \lambda_0^0 \, [0] + \lambda_0^1 \, [1] + \lambda_0^2 \, [2] + \lambda_0^3 \, [3]$$

[1] We say *formal* because we have not defined what the "sum of oriented faces" is. This can be done in full rigor, paying the price of abstraction. See the very first pages of [Mun84] for details.

[2] A clique is a subset of vertices of a graph such that every two vertices in the clique are adjacent.

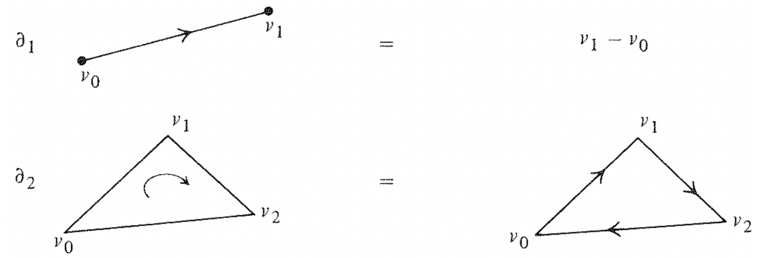


Figure 1: Effect of boundary map, Eq. (2), on elementary chains. From [Mun84, p. 29].

$$
\begin{aligned}
u_1 &= \lambda_1^{01}\,[01] + \lambda_1^{02}\,[02] + \cdots + \lambda_1^{23}\,[23] \\
u_2 &= \lambda_2^{012}\,[012] + \lambda_2^{123}\,[123]
\end{aligned}
$$

where $\lambda_k^e$ are real numbers.

**Boundary operator** Given the family of vector spaces $C_k$ it is natural to look for homomorphisms (also called linear maps or linear operators) between them.

**Definition 1.3.** Define the linear map

$$
\boxed{
\begin{aligned}
&\partial_k : C_k \to C_{k-1} \\
&\qquad [v_0 \dots v_k] \longmapsto \sum_{i=0}^{k} (-1)^k\,[v_0 \dots \hat{v}_i \dots v_k]
\end{aligned}
}
\tag{2}
$$

where the symbol $\hat{v}_i$ means that the vertex $v_i$ is to be deleted from the array. This map is called *boundary operator*.

The boundary of a $k$-chain is a $(k-1)$-chain: intuitively, the boundary of a segment is the difference between its vertices; the boundary of a triangle is a combination of its edges; and so on (see [Mun84, p. 29] for details), and Figure 1 for a geometrical intuition.

**Example 1.4.** Consider the same complex of Example (1.2). The boundary map acts on the elementary 2-chain $[012] \in B_2$ as

$$
\begin{aligned}
&\partial_2 : C_2 \to C_1 \\
&\qquad [012] \longmapsto [12] - [02] + [01]
\end{aligned}
$$

**Chain complex** Crucially, the boundary map fulfills

$$
\partial_k \circ \partial_{k+1} : C_{k+1} \to C_k \to C_{k-1} = 0 \tag{3}
$$

that is, the boundary of a boundary vanishes identically (the proof is a straightforward direct computation). Equivalently,

$$\operatorname{Im} \partial_{k+1} \subseteq \ker \partial_k \tag{4}$$

This is the main ingredient of a very general construction:

**Definition 1.5.** A *chain complex* is an algebraic structure that consists of a sequence of abelian groups and a sequence of homomorphisms between consecutive groups such that the image of each homomorphism is included in the kernel of the next.

Vector spaces are in particular abelian groups, so from a simplicial complex we can build the following chain complex:

$$\cdots \to C_{k+1} \xrightarrow{\partial_{k+1}} C_k \xrightarrow{\partial_k} C_{k-1} \to \ldots$$

In particular, we will focus on

$$C_2 \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0$$

The reason for this is that

- Hodge theorem says that each chain group $C_k$ can be decomposed as the direct sum of three special subspaces, the objects of our interest;
- The decomposition of each $C_k$ depends on $\partial_{k+1}$ and $\partial_k$;
- As we will see in Section (2), a game can be mapped to an object living in $C_1$, that is a 1-chain, or a *flow*.

Our next goals are then to

- understand the Hodge decomposition of each chain group;
- understand how to map a normal form game to $C_1$, perform the decomposition, and go back to the space of games;
- get more concrete, representing the boundary map (and the other linear maps we will introduce in the next section) as a matrix, and actually do computations.

To state Hodge theorem we need to introduce two more ingredients: the *coboundary operator* and the *Laplacian*.

## 1.2 Cochain complex

**Coboundary operator** The boundary operators allows us to go from the $k$-th chain group to the $(k-1)$-th one. Since we are in the realm linear algebra there is a natural way to obtain a map going "in the opposite direction":

**Definition 1.6.** The *coboundary operator*

$$d_{k-1} := \partial_k^* : C_{k-1}^* \to C_k^* \tag{5}$$

is the dual of the boundary operator $\partial_k : C_k \to C_{k-1}$.

For the general definition of dual of a vector space and of dual of a linear map between vector spaces see Appendix (A), but for practical uses it suffices to recall that

**Lemma 1.7.** *Let $A : V \to W$ be a linear map between vector spaces, and let $A^* : V^* \to W^*$ be its dual. The matrix representing $A^*$ is the transpose of the matrix representing $A$.*

**Corollary 1.8.** *The coboundary map fulfills*

$$d_k \circ d_{k-1} : C_{k-1}^* \to C_k^* \to C_{k+1}^* = 0 \tag{6}$$

*or equivalently*

$$\operatorname{Im} d_{k-1} \subseteq \ker d_k \tag{7}$$

*In particular, also the coboundary maps allow to build a chain complex:*

$$\cdots \leftarrow C_{k+1}^* \xleftarrow{d_k} C_k^* \xleftarrow{d_{k-1}} C_{k-1}^* \leftarrow \ldots$$

*Proof.* Immediate from Eq. (3) and Lemma (1.7). □

The chain complex built on the dual chain groups using the coboundary maps is called *cochain complex*. In the following we will restrict our attention to the piece of the chain and cochain complex involving 0-, 1- and 2-chains:

$$\begin{tikzcd}
C_0^* \arrow[r, bend left, "d_0=\partial_1^*"] & C_1^* \arrow[r, bend left, "d_1=\partial_0^*"] & C_2^* \\
C_0 & C_1 \arrow[l, "\partial_1"'] & C_2 \arrow[l, "\partial_2"']
\end{tikzcd} \tag{8}$$

**Constructions on inner product spaces** Note that all constructions performed so far are canonical: given a simplicial complex $K$, the chain and cochain complexes (that is, the chain groups, the boundary operator, and their dual) are naturally defined.

The next step, on the other hand, depends on a *choice*: each chain space $C_k$ is endowed with an inner product

$$\langle \cdot, \cdot \rangle_k : C_k \times C_k \to \mathbb{R} \tag{9}$$

that is a bilinear form symmetric and positive-definite.

This is a crucial choice, since the Hodge decomposition of each chain group will turn out to depend on these inner products. The inner product employed in [Can+11] is the Euclidean one: given two $k$-chains

$$u_k = \sum_{[e] \in B_k} \alpha_k^e \, [e] \in C_k$$

$$v_k = \sum_{[e] \in B_k} \beta_k^e \, [e] \in C_k$$

their Euclidean inner product is

$$\langle u_k, v_k \rangle_k^{\mathrm{eu}} = \sum_{[e] \in B_k} \alpha_k^e \, \beta_k^e \tag{10}$$

A vector space with an inner product is called *inner product space*.

*The reader interested in concrete applications in the Euclidean setting can safely skip section* (1.2.1) *and jump to section* (1.2.2)*.*

### 1.2.1 Non-Euclidean setting

We briefly review two constructions on inner product spaces: the identification between the space and its dual; and the adjoint of a linear map.

**Dual isomorphism** An inner product $\langle \cdot, \cdot \rangle : V \times V \to V^*$ induces an isomorphism between an inner product space $V$ and its dual $V^*$:

$$\langle \cdot, \cdot \rangle : V \overset{\sim}{\leftrightarrow} V^* \tag{11}$$

For details see Appendix (A); for our application it suffices to know that such isomorphism exists, and that in the case of the Euclidean inner product the matrix representing this isomorphism is the *identity matrix*, allowing for effectively identifying a vector space with its dual.

**Adjoint map** Given a linear map $A : V \to W$ between inner product spaces one can define the *adjoint map* $A^\dagger : W \to V$ by

$$\langle Av, w \rangle_W = \langle v, A^\dagger w \rangle_V, \quad \forall v \in V, w \in W$$

Let's remark that, given a linear map $A : V \to W$ between vector space, its dual map $A^* : W^* \to V^*$ is canonical; on the other hand its adjoint map $A^\dagger : W \to V$ depends on the inner products on $V$ and $W$. Remarkably, the dual map and the adjoint map are related by the isomorphism (11):

**Lemma 1.9.** *Given the linear map between inner product spaces $A : V \to W$, the dual map $A^*$ and the adjoint map $A^\dagger$ are related by*

$$A^\dagger = \langle \cdot, \cdot \rangle_V \circ A^* \circ \langle \cdot, \cdot \rangle_W$$

*that is, the following diagram commutes:*

$$\begin{tikzcd}
V \arrow[d, leftrightarrow, "\langle \cdot,\cdot \rangle_V"'] & W \arrow[l, "A^\dagger"'] \arrow[d, leftrightarrow, "\langle \cdot,\cdot \rangle_W"] \\
V^* & W^* \arrow[l, "A^*"]
\end{tikzcd}$$

*In particular, in the case of the Euclidean metric, the dual map $A^*$ and the adjoint map $A^\dagger$ are represented by the same matrix, that is the transpose of the matrix of $A$.*

In the diagram above (and in the following one) black maps are canonical, blue ones represent the isomorphisms induced by inner products; and red maps depend on these inner products.

With these notions at hand we can use the inner products defined in Eq. (9) to build adjoint maps on diagram (8):

$$\begin{tikzcd}
C_0^* \arrow[r, bend left, "d_0 = \partial_1^*"] \arrow[d, leftrightarrow, "\langle \cdot,\cdot \rangle_0"'] & C_1^* \arrow[l, bend left, "d_0^\dagger"] \arrow[r, bend left, "d_1 = \partial_0^*"] \arrow[d, leftrightarrow, "\langle \cdot,\cdot \rangle_1"'] & C_2^* \arrow[l, bend left, "d_1^\dagger"] \arrow[d, leftrightarrow, "\langle \cdot,\cdot \rangle_2"'] \\
C_0 & C_1 \arrow[l, "\partial_1"] & C_2 \arrow[l, "\partial_2"]
\end{tikzcd}$$

### 1.2.2 Euclidan setting

In the case of the Euclidean inner product we can forget about the difference between a space and its dual, and about the difference between an adjoint map and a dual map:

$$\begin{tikzcd}
C_0 \arrow[r, bend left, "d_0 = \partial_1^* = \partial_1^\dagger"] & C_1 \arrow[l, bend left, "\partial_1 = d_0^* = d_0^\dagger"] \arrow[r, bend left, "d_1 = \partial_0^* = \partial_0^\dagger"] & C_2 \arrow[l, bend left, "\partial_2 = d_1^* = d_1^\dagger"]
\end{tikzcd} \quad \text{[Euclidean]} \tag{12}$$

Concretely, the maps in the lower row are defined by eq. (2), and the maps in the upper row are determined by Lemma (1.7).

**Laplacian** The last ingredient we need for Hodge theorem is the *Laplacian operator*; we define it only on $C_1$ and in the Euclidean case, but it is defined analogously on chain groups of any rank and for arbitrary inner products.

**Definition 1.10.** The *Laplacian* operator is

$$\Delta_1 := d_0 \circ \partial_1 + \partial_2 \circ d_1 : C_1 \to C_1 \tag{13}$$

This leaves us with the following diagram:

$$\begin{tikzcd} C_0 \arrow[r, bend left, "d_0"] & C_1 \arrow[l, bend left, "\partial_1"] \arrow[r, bend left, "d_1"] \arrow[loop above, "\Delta_1"] & C_2 \arrow[l, bend left, "\partial_2"] \end{tikzcd} \quad \text{[Euclidean]} \tag{14}$$

In the following we consider always the Euclidean setting.

## 1.3 Combinatorial Hodge theorem

Recall by Eqs. (4) and (7) that

$$\begin{aligned} \operatorname{Im} d_0 &\subseteq \ker d_1 \\ \operatorname{Im} \partial_2 &\subseteq \ker \partial_1 \end{aligned} \tag{15}$$

Furthermore it is a standard result that

$$\ker \Delta_1 = \ker \delta_1 \cap \ker d_1 \tag{16}$$

These vector subspaces of $C_1$ deserve special names:

**Definition 1.11.** A 1-chain in $C_1$ is called

- *exact* iff it belongs to $\operatorname{Im} d_0$
- *closed* iff it belongs to $\ker d_1$
- *co-exact* iff it belongs to $\operatorname{Im} \partial_2$
- *co-closed* iff it belongs to $\ker \partial_1$
- *harmonic* iff it belongs to $\ker \Delta_1$

Note by Eq. (15) that any (co-)exact chain is (co-)closed, and by Eq. (16) that a 1-chain is harmonic iff it is closed and co-closed. In the following we use interchangeably the terms 1-*chain* and *flow* for elements of $C_1$.

**Theorem 1.12** (Hodge). *The chain group $C_1$ admits the orthogonal decomposition*

$$\begin{aligned} C_1 &= \textit{closed} \oplus \textit{closed}^{\perp} \\ &= \textit{exact} \oplus \textit{harmonic} \oplus \textit{co-exact} \end{aligned} \tag{17}$$

See Appendix (B.1) for a proof sketch. The first line holds true by definition of orthogonal complement, and it is a general fact that

$$\text{closed}^{\perp} = (\ker d_1)^{\perp} = \operatorname{Im} d_1^* = \operatorname{Im} d_1^* \operatorname{Im} \partial_2 = \text{co-exact}$$

The real content of Hodge theorem, and what we will actually need, is that closed flows admit the orthogonal decomposition

$$\text{closed} = \text{exact} \oplus \text{harmonic} \tag{18}$$

## 2 Deviation Map of Normal Form Games

The goal of this section is to represent a normal form game as a 1-chain living in $C_1$ on some graph - the *response graph* - , so that we can apply Hodge theorem. This will be done by a so-called *deviation map* $D$ acting as

$$D : \text{ space of games} \to C_1$$

In the next sections we build the relevant graph, define what we mean by *space of games*, and define the deviation map $D$ in Eq. (23).

**Normal form game** A normal form game is a tuple $\Gamma = (\mathcal{N}, \mathcal{A}, u)$ where

- $\mathcal{N} = \{1, 2, \ldots, N\}$ is the set of players
- Each player $i \in \mathcal{N}$ has a set of pure strategies
$$\mathcal{A}_i = \{1, 2, \ldots, A_i\}$$
- $\mathcal{A} = \prod_{i \in \mathcal{N}} \mathcal{A}_N$ is the set of pure strategy profiles
- Each player has an individual utility function
$$u_i : \mathcal{A} \to \mathbb{R}, \quad a \mapsto u_i(a)$$
- The *utility map of the game* is
$$u : \mathcal{A} \to \mathbb{R}^N, \quad a \mapsto (u_1, \ldots, u_N)(a)$$

The number of players is $N = |\mathcal{N}|$, the number of pure strategies of player $i \in \mathcal{N}$ is $A_i = |\mathcal{A}_i|$, and the number of pure strategies profiles is $A = |\mathcal{A}| = \prod_{i \in \mathcal{N}} A_i$, so the number of utilities is $AN$.

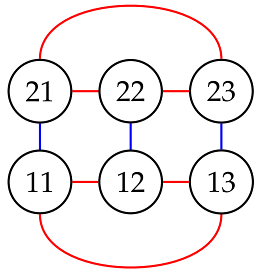


Figure 2: Response graph of a $[2,3]$ game as in Example (2.1). Blue (vertical) edges represent deviations of the first player and red (horizontal and bent) edges represent deviations of the second player.

**Game skeleton** We will call the datum of a set of players and of a space of pure strategies for each player, without specifying a utility map, the *skeleton* of a normal form game, and denote it by $(\mathcal{N}, \mathcal{A})$ or by $[A_1, A_2, \ldots, A_N]$. For example $[3,4]$ is the skeleton of a normal form game with 2 players, where player 1 has 3 pure strategies and player 2 has 4 pure strategies.

**Unilateral deviations** Pairs of strategy profiles $a, b \in \mathcal{A}$ that differ only in the strategy of one player are called *unilateral deviations*, and their space is denoted by $\mathcal{E}$. One can check that given the skeleton of a normal form game $(\mathcal{N}, \mathcal{A})$ the number of unilateral deviations is

$$E := |\mathcal{E}| = \frac{A}{2} \sum_{i \in \mathcal{N}} (A_i - 1) \tag{19}$$

**Response graph** This allows to draw a graph from the skeleton $(\mathcal{N}, \mathcal{A})$ of a normal form game by drawing a node for each pure strategy profile in $\mathcal{A}$, and an edge for each unilateral deviation in $\mathcal{E}$. This gives a graph with $A$ nodes and $E$ edges. In the language of section (1.1) (in particular cf. Example (1.2))

- $\mathcal{A}$ is the basis of the chain group $C_0$
- $\mathcal{E}$ is the basis of the chain group $C_1$

**Example 2.1.** Consider a game with skeleton $[2,3]$:

- $\mathcal{N} = \{1,2\}, \quad N = 2$
- $\mathcal{A}_1 = \{1,2\}, \mathcal{A}_2 = \{1,2,3\}$

The space of pure strategy profiles, with cardinality $A = 6$, is

$$\text{Basis of } C_0 = \mathcal{A} = \{(1,1), (1,2), (1,3), (2,1), (2,2), (2,3)\} \tag{20}$$

The space of unilateral deviations, with cardinality $E = 9$, is

$$\begin{aligned} \text{Basis of } C_1 = \mathcal{E} = \{ &[(1,1),(1,2)], [(1,1),(1,3)], [(1,1),(2,1)], \\ &[(1,2),(1,3)], [(1,2),(2,2)], [(1,3),(2,3)], \\ &[(2,1),(2,2)], [(2,1),(2,3)], [(2,2),(2,3)] \} \end{aligned} \tag{21}$$

The corresponding response graph is shown in Figure 2.

## 2.1 Matrix representation of $\partial_1$ and $d_0$

Having an explicit basis for $C_1$ and $C_0$ allows for an explicit representation of the matrices of the boundary and co-boundary by Eq. (2) and Lemma (1.7).

Since we're dealing with linear maps we only need to look at how they act on basis vectors, so to write the $(A \times E)$-dimensional matrix of $\partial_1 : C_1 \to C_0$ let's look at how it acts on $\mathcal{E}$.

For a game with skeleton $[2,3]$ ordering the basis $\mathcal{A}$ and $\mathcal{E}$ as in Example (2.1) yields

$$\begin{aligned} \partial_1 : C_1 &\to C_0 \\ [(1,1),(1,2)] &\longmapsto (1,2) - (1,1) \\ [(1,1),(1,3)] &\longmapsto (1,3) - (1,1) \\ &\vdots \\ [(2,2),(2,3)] &\longmapsto (2,3) - (2,2) \end{aligned}$$

corresponding to the $(6 \times 9)$ matrix

$$[\partial_1] = \begin{pmatrix} -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Some guesswork will show that this algorithm works for arbitrary skeletons:

```
    def make_boundary_1_matrix(self):
        """Matrix of partial_1: C_1 --> C_0"""

        # Start with transpose
        A = np.zeros([int(self.dim_C1), int(self.dim_C0)])

        for row in range(int(self.dim_C1)):
            basis_edge = self.edges[row]

            minus_node, plus_node = basis_edge
            minus_column = self.nodes.index( minus_node )
```

```
            plus_column = self.nodes.index( plus_node )
            A[row][minus_column] = -1
            A[row][plus_column] = +1

        return A.transpose()
```

Once the matrix representing $\partial_1 : C_1 \to C_0$ is known, the matrix representing $d_0 : C_0 \to C_1$ is obtained for free as its transpose by Lemma (1.7).

## 2.2 Vector space of games and deviation map

**Vector space of games** When we refer to a *game* we usually mean a utility map given a fixed skeleton. In other words, the space of games $U$ is the space of possible utility maps $u$ given a skeleton $(\mathcal{N}, \mathcal{A})$. This is the assignment of $N$ utilities (one per player) to $A$ pure strategy profiles, that is an $AN$-dimensional vector space.

Recall that, given a graph, $C_0$ is the space of possible assignments of one number to each node in the graph. Since by definition the space of games is the space of possible assignments of $N$ numbers to each node in the response graph, the space of games is the Cartesian product of $N$ copies of $C_0$:

$$U = \underbrace{C_0 \times \cdots \times C_0}_{N \text{ times}}$$

A basis for $U$ is then given by the disjoint union of $N$ copies of $\mathcal{A}$.

**Example 2.2.** Consider a game with skeleton $[2,3]$ as in Example (2.1), and consider the following payoff map:

$$\begin{aligned} u :\ & \mathcal{A} \to \mathbb{R}^2 \\ & (1,1) \longmapsto (-3,3) \\ & (1,2) \longmapsto (0,-5) \\ & (1,3) \longmapsto (-3,3) \\ & (2,1) \longmapsto (3,0) \\ & (2,2) \longmapsto (-3,0) \\ & (2,3) \longmapsto (0,1) \end{aligned} \quad \xleftrightarrow{\text{bimatrix notation}} \quad \begin{pmatrix} -3,3 & 0,-5 & -3,3 \\ 3,0 & -3,0 & 0,1 \end{pmatrix}$$

The dimension of $U$ is $AN = 12$, and a basis is given by

$$\begin{aligned} \text{Basis of } U = \mathcal{A} \sqcup \mathcal{A} = \{ & (1,1)_1, (1,2)_1, (1,3)_1, (2,1)_1, (2,2)_1, (2,3)_1, \\ & (1,1)_2, (1,2)_2, (1,3)_2, (2,1)_2, (2,2)_2, (2,3)_2 \} \end{aligned} \tag{22}$$

The payoff map $u$ can be seen as the vector

$$u = -3\,(1,1)_1 + 0\,(1,2)_1 + \cdots + 0\,(2,2)_2 + 1(2,3)_2 \in U$$

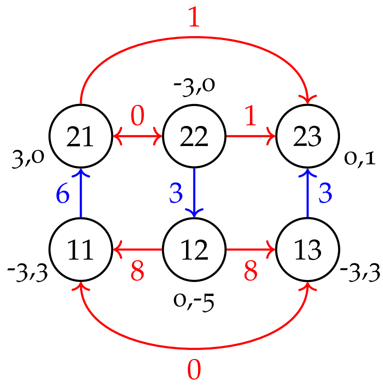


Figure 3: Deviation flow of the game described in Example (2.2).

**Deviation map** Given a game skeleton $(\mathcal{N}, \mathcal{A})$ we are finally in the position to map a game - that is, a vector $u \in U$ - to a 1-chain, or *flow*, that is a vector in the first chain group $C_1$ of the response graph built from the skeleton $(\mathcal{N}, \mathcal{A})$.

Recall that a flow is the assignment of a number to each edge of the response graph. A natural way to make these assignments given a utility map $u$ is to label each edge by the utility difference of the deviating player:

**Definition 2.3.**

$$\boxed{\begin{aligned} D : U &\to C_1 \\ u &\longmapsto Du \end{aligned} \quad \text{s.t.} \ \forall [a,b] \in \mathcal{E} \quad (Du)_{ab} = u_i(b) - u_i(a) \\ \text{for } i \text{ s.t. } a_i \neq b_i} \tag{23}$$

The map $D$ is linear, and is called *deviation map*. Given a game $u \in U$, the flow $Du \in C_1$ is called *deviation flow of the game*.

In this definition $[a,b]$ is an edge in the response graph and $(Du)_{ab}$ is the number the flow $Du$ assigns to such edge. By convention, the edges in the response graph are oriented such that $(Du)_{ab} \geq 0$ for all $[a,b] \in \mathcal{E}$.

The deviation flow of a game captures its strategic structure: the orientation of the edges of the response graph reflects the interest of each player at each state of the game. If an arrow leaves a node, a player following the arrow does not lose.

**Example 2.4.** Consider the normal form game of Example (2.2), and recall that its response graph is shown in Figure 2. The deviation flow of this game is shown in Figure 3.

## 2.3 Matrix representation of $D$

Having an explicit basis for $U$ and $C_1$ one can write the matrix representing the deviation map $D : U \to C_1$, of dimension $(E \times AN)$.

**Example 2.5.** Consider the usual game with skeleton $[2, 3]$ as in Example (2.1); recall that in this case $N = 2$, $A = 6$, and $E = 9$. With basis for $C_1$ and $U$ given respectively by Eq. (21) and Eq. (22) one gets the $(9 \times 12)$ matrix

$$[D] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

For arbitrary skeletons, the following algorithm does the trick:

```

    self.payoff_basis = [ (i.player_name-1, a) for i in self.players for
    a in self.strategy_profiles ]

    def make_pwc_matrix(self):
        """Matrix of deviation map: U --> C_1"""
        A = np.zeros([int(self.dim_C1), int(self.dim_C0N)])

        for row in range(int(self.dim_C1)):
            edge = self.edges[row]
            i = utils.different_index(edge)

            minus_column = self.payoff_basis.index( (i, edge[0]) )
            plus_column = self.payoff_basis.index( (i, edge[1]) )
            A[row][minus_column] = -1
            A[row][plus_column] = +1

        return np.asmatrix(A)
```

# 3 Hodge Decomposition of Normal Form Games

We can finally state our decomposition theorem for normal form games, first in an abstract form (up to a normalization choice), then in a concrete choice that allows for computation (after a suitable choice of normalization).

**Definition 3.1.** Consider the game skeleton $(\mathcal{N}, \mathcal{A})$ and the induced deviation map

$D : U \to C_1$. The space of *non-strategic games* is the kernel of the deviation map:

$$\mathcal{K} := \ker D \subseteq U \tag{24}$$

**Theorem 3.2** (Abstract Decomposition Theorem)**.** *Consider the game skeleton* $(\mathcal{N}, \mathcal{A})$*. The space of games* $U$ *admits the orthogonal decomposition*

$$U \cong \mathcal{K} \oplus \textit{exact flows} \oplus \textit{harmonic flows} \tag{25}$$

Note that in this theorem the exact flows and the harmonic flows live in $C_1$, while $\mathcal{K}$ lives in the space of games $U$. After making a normalization choice we will state the theorem with an equality, not an isomorphism, i.e. decompose $U$ as the orthogonal direct sum of three subspaces.

To prove this theorem we need the following lemma.

### 3.1 Feasible flows are precisely closed flows

**Lemma 3.3.** *Consider the game skeleton* $(\mathcal{N}, \mathcal{A})$ *and the induced deviation map* $D : U \to C_1$*. The image of the deviation map is precisely the space of closed flows:*

$$\operatorname{Im} D = \ker d_1 \subseteq C_1 \tag{26}$$

The image of the deviation map is the space of *feasible flows*, that is the subspace of $C_1$ containing those flows that can be realized on the response graph of a normal form game; and the content of this lemma is that these are precisely the *closed flows*.

**What is a closed flow?** By definition a flow $X \in C_1$ is closed iff $X \in \ker d_1$, that is iff $d_1 X = 0$. One can obtain an explicit expression for the co-boundary operator $d_1 : C_1 \to C_2$ analogue to that of the boundary operator $\partial_k : C_k \to C_{k-1}$ given in Eq. (2) (see e.g. Def. 4.2 in [Jia+11]), and it turns out that a flow $X \in C_1$ is closed iff

$$X_{ab} + X_{bc} + X_{cd} = 0$$

for any 3-clique $[abc]$ in the response graph, where $X_{ab}$ is the number the flow $X$ assigns to the edge $[ab]$.

This allows to get an idea of what this Lemma means.

*Sketch proof of Lemma* (3.3). The fact that $\operatorname{Im} D \subseteq \ker d_1$ is stated in [Can+11], but it is proved employing a relatively heavy machinery, while we developed a simpler argument. Let $u \in U$, then

$$\begin{aligned}
(d_1 D u)\,(abc) &= (Du)_{ab} + (Du)_{bc} + (Du)_{ca} \\
&= u_i(b) - u_i(a) + u_j(c) - u_j(b) + u_h(a) - u_h(c) \\
&= 0 \text{ since } (abc) \text{ is a 3-clique} \Rightarrow i = j = h \quad \square
\end{aligned}$$

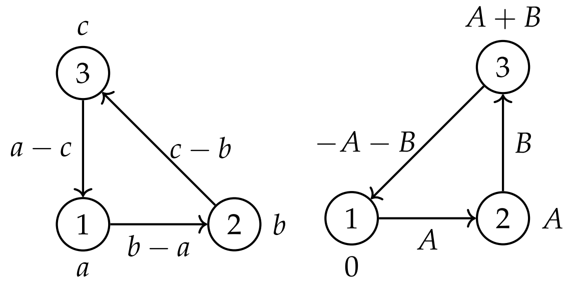


Figure 4: Left: every feasible flow is a closed flow. Right: every closed flow is feasible. See Ex. (3.4) for details.

This means that $d_1 \circ D \equiv 0$, i.e. every feasible flow is a closed flow.

The proof of the fact that $\operatorname{Im} D \supseteq \ker d_1$ is, to our knowledge, original. Given a closed flow $X \in C_1$ we need to find a utility $u$ such that $Du = X$. The idea is to factorize the response graph into complete sub-graphs that have a unique deviating player, and to decouple the system of equations $Du_{ab} = X_{ab}$ into sub-systems relative to these sub-graphs. With this decomposition in place the problem is reduced to showing that if $X$ is closed than it is exact on each complete sub-graph. This is true by Poincarè lemma [DLM05], since each complete sub-graph is contractible (seeing the response graph as a 2-dimensional simplicial complex). □

**Example 3.4.** See Figure 4.

Left : $\operatorname{Im} D \subseteq \ker d_1$

$$\begin{aligned}
&(d_1 Du)(abc) = \\
&= Du(ab) + Du(bc) + Du(ca) \\
&= b - a + c - b + a - c = 0
\end{aligned}$$

Right : $\operatorname{Im} D \supseteq \ker d_1$

$$\begin{aligned}
&0 = (d_1 X)(abc) = \\
&= X(ab) + X(bc) + X(ca) \\
&\Rightarrow \exists u : Du = X
\end{aligned}$$

## 3.2 Proof of the Abstract Decomposition Theorem

With this result at hand, the proof of our abstract decomposition theorem (3.2) is immediate.

*Proof of Theorem* (3.2). Since $D : U \to C_1$ is a linear map it holds true that

$$U \cong \ker D \oplus \operatorname{Im} D$$

and we are done already by Lemma (3.3) and Eq. (18):

$$\operatorname{Im} D = \text{closed flows} = \text{exact flows} \oplus \text{harmonic flows}$$

$$U \cong \mathcal{K} \oplus \text{exact flows} \oplus \text{harmonic flows}$$

$\square$

## 3.3 Normalized Decomposition Theorem

To make the decomposition concrete one can choose *any* complement $\bar{\mathcal{K}}$ of the space of non-strategic games $\mathcal{K}$ in $U$. Indeed for any choice of $\bar{K}$ such that $U = \mathcal{K} \oplus \bar{\mathcal{K}}$ it holds true that $\bar{\mathcal{K}}$ decomposes in $U$ as

$$\bar{\mathcal{K}} = \left(D^{-1}(\text{exact flows}) \cap \bar{\mathcal{K}}\right) \oplus \left(D^{-1}(\text{harmonic}) \cap \bar{\mathcal{K}}\right) \tag{27}$$

See Appendix (B.2) for details.

**Definition 3.5.** Consider the game skeleton $(\mathcal{N}, \mathcal{A})$ and the induced deviation map $D : U \to C_1$.

The space of *normalized potential games* is

$$\mathcal{P} := \left(D^{-1}(\text{exact flows}) \cap \bar{\mathcal{K}}\right) \equiv \left(D^{-1}(\operatorname{Im} d_0) \cap \bar{\mathcal{K}}\right) \subseteq U \tag{28}$$

The space of *normalized harmonic games* is

$$\mathcal{H} := \left(D^{-1}(\text{harmonic flows}) \cap \bar{\mathcal{K}}\right) \equiv \left(D^{-1}(\ker \Delta_1) \cap \bar{\mathcal{K}}\right) \subseteq U \tag{29}$$

The space of *potential games* is

$$D^{-1}(\text{exact flows}) \equiv D^{-1}(\operatorname{Im} d_0) \subseteq U \tag{30}$$

The space of *harmonic games* is

$$D^{-1}(\text{harmonic flows}) \equiv D^{-1}(\ker \Delta_1) \subseteq U \tag{31}$$

**Lemma 3.6.**

$$\textit{potential games} = \mathcal{P} \oplus \mathcal{K} \tag{32}$$

$$\textit{harmonic games} = \mathcal{H} \oplus \mathcal{K} \tag{33}$$

We refer to a choice of a complement $\bar{\mathcal{K}}$ for the kernel $\mathcal{K}$ of the deviation map $D$ as *normalization*. Given a normalization we have the Normalized Decomposition Theorem:

**Theorem 3.7** (Normalized Decomposition Theorem)**.** *Consider the game skeleton $(\mathcal{N}, \mathcal{A})$. The space of games $U$ admits the orthogonal decomposition*

$$U = \mathcal{P} \oplus \mathcal{K} \oplus \mathcal{H} \tag{34}$$

*Proof.* By definition of complement $U = \mathcal{K} \oplus \bar{\mathcal{K}}$, and the conclusion follows from Eq. (27) and by definition of $\mathcal{P}$ and $\mathcal{H}$. □

**Harmonic games** An important consequence of Lemma (3.3) is that harmonic games just depend on $\partial_1$ (and not also on $d_1$, as one would expect). Recall by Eq. (16) that

$$\text{harmonic flows} = \ker \Delta_1 = \ker \partial_1 \cap \ker d_1$$

Now since $\operatorname{Im} D = \ker d_1$ we have that

$$\operatorname{Im} D \cap \ker \Delta_1 = \operatorname{Im} D \cap \ker \partial_1 \Rightarrow D^{-1}(\ker \Delta_1) = D^{-1}(\ker \partial_1)$$

So by definition of harmonic games we have that

**Lemma 3.8.** *The space of normalized harmonic games is*

$$\mathcal{H} = \left(D^{-1}(\ker \partial_1) \cap \bar{\mathcal{K}}\right) \subseteq U \tag{35}$$

This is computationally very convenient because the matrix representation of $\partial_1 : C_1 \to C_0$ is relatively inexpensive to find (c.f. Sec. (2.1)), while the matrix representation of $d_1 : C_1 \to C_2$ involves finding the 3-cliques of the response graph, which is computationally more expensive for big graphs.

## 3.4 Orthogonal Normalization and Moore-Penrose Inverse

In [Can+11] the authors choose to use the *Euclidean orthogonal* complement of $\mathcal{K}$:

$$\bar{\mathcal{K}} \stackrel{!}{=} \mathcal{K}^{\perp} \tag{36}$$

With this choice, the decomposition of a game can be performed explicitly as follows.

By definition of $U = \mathcal{K} \oplus \mathcal{K}^{\perp}$ any $u \in U$ admits the unique decomposition

$$u = u_{\mathcal{K}} + \pi u \tag{37}$$

where

$$\pi : U \to U \tag{38}$$

is the orthogonal projection onto $\operatorname{Im} \pi = \mathcal{K}^{\perp}$ along $\ker \pi = \mathcal{K}$.

This provides the non-strategic part of the decomposition. To get the potential and the harmonic part, map $D : u \mapsto Du \in \operatorname{Im} D = \text{closed flows} \subseteq C_1$. By Hodge theorem $Du$ can then be decomposed as

$$Du = \mathrm{e}\, Du + (1 - \mathrm{e})\, Du \in \ker d_1 \tag{39}$$

where

$$\mathrm{e} : C_1 \to C_1 \tag{40}$$

is the orthogonal projection onto $\operatorname{Im} \mathrm{e} = \text{exact flows}$ along $\ker \mathrm{e} = (\text{exact flows})^\perp$.

**Moore-Penrose Inverse** The Moore-Penrose Inverse provides a convenient way to compute these projection operators.

Given a linear map $A : V \to W$ between vector spaces endowed with the Euclidean inner product there exists a unique linear map $\tilde{A} : W \to V$ fulfilling some properties (see [KS05; RAG08]); what we care about here is that

$$\tilde{A}A : V \to V \tag{41}$$

is the orthogonal projection

- onto $\operatorname{Im} \tilde{A}A = \operatorname{Im} \tilde{A} = (\ker A)^\perp$
- along $\ker \tilde{A}A = \ker A$

and analogously

$$A\tilde{A} : W \to W \tag{42}$$

is the orthogonal projection

- onto $\operatorname{Im} A\tilde{A} = \operatorname{Im} A$
- along $\ker A\tilde{A} = \ker \tilde{A} = (\operatorname{Im} A)^\perp$

If $A$ is bijective then $\tilde{A} = A^{-1}$ is just the inverse of $A$; convince yourself that in this case the properties above are true.

**Projection operators** to detail Recall that $D : U \to C_1$ and $d_0 : C_0 \to C_1$. With this at hand we can build the projection operator $\pi$ of Eq. (38) as

$$\pi = \tilde{D}D : U \to U \tag{43}$$

that is the orthogonal projection

- onto $\operatorname{Im} \tilde{D}D = \operatorname{Im} \tilde{D} = (\ker D)^\perp = \mathcal{K}^\perp$
- along $\ker \tilde{D}D = \ker D = \mathcal{K}$

as we need. Analogously we can build the projection operator e of Eq. (40) as

$$e = d_0 \tilde{d}_0 : C_1 \to C_1 \tag{44}$$

that is the orthogonal projection

- onto $\operatorname{Im} d_0 \tilde{d}_0 = \operatorname{Im} d_0 =$ exact flows
- along $\ker d_0 \tilde{d}_0 = \ker \tilde{d}_0 = (\operatorname{Im} d_0)^{\perp} = (\text{exact flows})^{\perp}$

as we need.

**The decomposition, finally** This gives an explicit receipt to decompose a utility function:

$$\boxed{\begin{aligned} u_{\mathcal{K}} &= u - \pi u \\ u_{\mathcal{P}} &= \tilde{D} e D u \\ u_{\mathcal{H}} &= u - u_{\mathcal{N}} - u_{\mathcal{P}} \end{aligned}} \tag{45}$$

This happens in the `Payoff.decompose_payoff()` method:

```
    def decompose_payoff(self):

        print('start decomposition')

        u = self.payoff_vector

        PI = self.game.normalization_projection
        e = self.game.exact_projection

        PWC_pinv = self.game.pwc_matrix_pinv
        PWC = self.game.pwc_matrix
        delta_0_pinv = self.game.coboundary_0_matrix_pinv
        delta_0 = self.game.coboundary_0_matrix

        print('this seems to be the bottleneck, big matrices
    multiplication')

        uN = u - PI @ u
        print('first multiplication done')

        uP = PWC_pinv @ e @ PWC @ u
        print('three more multiplications done, this is slowest step!')

        uH = u - uN - uP
```

The projection operators `game.normalization_projection` and `game.exact_projection` are computed in the `Game` class:

```
        # Pseudo-Inverse and projection block
        # Moore-Penrose pseudo-inverse of pwc
        print('start PINV block')
        self.pwc_matrix_pinv = npla.pinv(self.pwc_matrix)
```

```
        # PI: CON --> CON projection onto Euclidean orthogonal complement of ker(D)
        self.normalization_projection = np.matmul(self.pwc_matrix_pinv, self.pwc_matrix)

        # pinv(d_0): C^1 --> C^0
        self.coboundary_0_matrix_pinv = npla.pinv(self.coboundary_0_matrix)

        # e: C1 --> C1 projection onto exact
        self.exact_projection = np.matmul(self.coboundary_0_matrix, self.coboundary_0_matrix_pinv)
```

**Potential function** The potential function relative to the potential component $u_{\mathcal{P}}$ is

$$\phi = \tilde{d_0} D u \in C_0 \tag{46}$$

with $u \in U$, $D : U \to C_1$, and $\tilde{d_0} : C_1 \to C_0$.

# A Dual and Adjoint in Linear Algebra

## A.1 Dual Homomorphism

See [RAG08]. Let $A$ and $G$ be abelian groups; for definiteness think of $A$ as a vector space and $G$ as the real numbers.

The *dual* of $A$ is $A^* := \mathrm{Hom}\,(A, G)$. This is an abelian group if we add two homomorphism in $A^*$ adding their value in $G$.

Let $B$ be another abelian group. A homomorphism $f : A \to B$ gives rise to a *dual homomorphism*

$$\begin{aligned} f^* : B^* &\to A^* \\ \phi &\longmapsto f^*\phi = \phi \circ f \end{aligned} \tag{47}$$

meaning that this diagram commutes

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[dr, "f^*\phi"'] & B \arrow[d, "\phi"] \\ & \mathbb{R} \end{tikzcd} \tag{48}$$

This construction is canonical.

## A.2 Dual Isomorphisms

Let's endow $A$ with an inner product

$$\langle \cdot, \cdot \rangle_A : A \times A \to \mathbb{R} \tag{49}$$

This induces a non-canonical isomorphisms $A \cong A^*$ by

$$\sharp_A : A^* \xrightarrow{\sim} A \tag{50}$$

such that

$$\phi(a) = \langle \phi^{\sharp_A}, a \rangle_A \in \mathbb{R}, \quad \forall \phi \in A^*, \forall a \in A \tag{51}$$

where the notation $\phi^{\sharp_A}$ is used for $\sharp_A(\phi)$. The inverse of $\sharp_A$ is denoted by

$$(\sharp_A)^{-1} = \flat_A : A \xrightarrow{\sim} A^* \tag{52}$$

hence

$$u^{\flat_A}(a) = \langle u, a \rangle_A \in \mathbb{R}, \quad \forall u, a \in A \tag{53}$$

Finally, an inner product on $A$ induces the inner product on $A^*$

$$\langle \cdot, \cdot \rangle_{A^*} : A^* \times A^* \to \mathbb{R} \tag{54}$$

such that

$$\langle \phi, \psi \rangle_{A^*} = \langle \phi^{\sharp_A}, \psi^{\sharp_A} \rangle_A \quad \forall \phi, \psi \in A^* \tag{55}$$

Analogously endowing $B$ with an inner product

$$\langle \cdot, \cdot \rangle_B : B \times B \to \mathbb{R} \tag{56}$$

one defines

$$\sharp_B : B^* \xrightarrow{\sim} B, \quad (\sharp_B)^{-1} = \flat_B : B \xrightarrow{\sim} B^*, \quad \langle \cdot, \cdot \rangle_{B^*} : B^* \times B^* \to \mathbb{R} \tag{57}$$

## A.3 Adjoint Homomorphism

Given inner products on $A$ and $B$ as above a homomorphism $f : A \to B$ gives rise to an *adjoint homomorphism*

$$f^\dagger : B \to A \tag{58}$$

such that

$$\langle fa, b \rangle_B = \langle a, f^\dagger b \rangle_A, \quad \forall a \in A, \forall b \in B \tag{59}$$

**Lemma A.1.** *The dual of $f$ and the adjoint of $f$ are related by*

$$f^\dagger = \sharp_A \circ f^* \circ \flat_B \tag{60}$$

*that is, the following diagram commutes:*

$$\begin{tikzcd} A & B \arrow[l, "f^\dagger"'] \arrow[d, "\flat_B"] \\ A^* \arrow[u, "\sharp_A"] & B^* \arrow[l, "f^*"] \end{tikzcd} \tag{61}$$

*Proof.* Take $a \in A, b \in B$. Then

$$\begin{aligned}\mathbb{R} \ni \langle f^{\dagger}b, a\rangle_A &= \langle b, fa\rangle_B \\ &= b^{\flat_B}(fa) = \left(b^{\flat_B} \circ f\right)(a) \\ &= (f^* b^{\flat_B})(a) \\ &= \langle \left(f^* b^{\flat_B}\right)^{\sharp_A}, a\rangle_A\end{aligned}$$

and we conclude by bilinearity and non-degeneracy of the inner product. □

# B Proofs

## B.1 Sketch Proof of Hodge Theorem

**Lemma B.1.**

$$A : V \to W$$
$$\frac{V}{\ker A} \cong \operatorname{Im} A$$

**Lemma B.2.** *For any subspace $S$ of $V$*

$$V \cong S \oplus \frac{V}{S}$$

**Lemma B.3.**

$$A : V \to W$$
$$V \cong \ker A \oplus \frac{V}{\ker A} \cong \ker A \oplus \operatorname{Im} A$$

**Lemma B.4.**

$$\textit{co-exact} = (\textit{closed})^{\perp}$$
$$\operatorname{Im}(d_1^{\dagger}) = (\ker d_1)^{\perp}$$

**Definition B.5.** - Cohomology group

$$H_1 := \frac{\text{closed}}{\text{exact}} = \frac{\ker d_1}{\operatorname{Im} d_0}$$

- Harmonic space

$$\mathcal{H}_1 = \{c \in C^1 : \Delta_1 c = 0\}$$

**Theorem B.6** (Hodge)**.** *The space of harmonic forms is isomorphic to the cohomology group*

$$H_1 \cong \mathcal{H}_1$$

**Corollary B.7** (Hodge)**.** *The cochain group $C^1$ decomposes (uniquely and orthogonally) as*

$$\begin{aligned} C^1 &= \textit{exact} \oplus \textit{co-exact} \oplus \textit{harmonic} \\ &= \operatorname{Im} d_0 \oplus \operatorname{Im} d_1^\dagger \oplus \ker \Delta_1 \end{aligned}$$

*Proof.*

$$C^1 = \ker d_1 \oplus (\ker d_1)^\perp = \ker d_1 \oplus \operatorname{Im} d_1^\dagger$$

First identity by fact that $V = S \oplus S^\perp$ for any subspace $S \subseteq V$ by definition of orthogonal projection; second by Lemma (4) on image of adjoint map.

So one piece is done: $\operatorname{Im} d_1^\dagger$ are co-exact.

It is left to show that

$$\text{closed} = \ker d_1 = \text{exact} \oplus \text{harmonic} = \operatorname{Im} d_0 \oplus \ker \Delta_1$$

Since exact $\subseteq$ closed we can quotient and by Lemma (2) above

$$\text{closed} \cong \text{exact} \oplus \frac{\text{closed}}{\text{exact}}$$

$$\ker d_1 \cong \operatorname{Im} d_0 \oplus \frac{\ker d_1}{\operatorname{Im} d_0}$$

So just with standard linear algebra we can get as far as

$$C^1 \cong \text{exact} \oplus \frac{\text{closed}}{\text{exact}} \oplus \text{co-exact}$$

The crucial step is now Hodge theorem: there is a unique way to choose a harmonic representative in each cohomology group. So

$$C^1 \cong \text{exact} \oplus \text{harmonic} \oplus \text{co-exact}$$

□

## B.2 Proof of Eq. (27)

**Lemma B.8.** *Given a linear map*

$$D : U \to C_1$$

*let*

$$\mathcal{K} = \ker D$$

*If*

$$\operatorname{Im} D = A \oplus B$$

*then for any complement $\bar{K}$ of $K$ in $U$ we have*

$$\bar{\mathcal{K}} = \left(D^{-1}(A) \cap \bar{\mathcal{K}}\right) \oplus \left(D^{-1}(B) \cap \bar{\mathcal{K}}\right)$$

*Proof.*

$$U = \mathcal{K} \oplus \bar{\mathcal{K}}$$

The restriction of $D$ to $\bar{K}$ gives a non-canonical isomorphism

$$D|_{\bar{K}} : \bar{K} \xrightarrow{\sim} \operatorname{Im} D$$

since for any $Du \in \operatorname{Im} D$ one can decompose $u = u_{\mathcal{K}} + u_{\bar{K}}$ with $u_{\bar{K}} \in \bar{K}$ and $Du = Du_{\bar{K}}$.

Now the claim is that any $u_{\bar{\mathcal{K}}}$ can be decomposed as $u_{\bar{\mathcal{K}}} = \alpha + \beta$ for some $\alpha \in D^{-1}(A) \cap \bar{K}$ and $\beta \in D^{-1}(B) \cap \bar{K}$. So given any $u_{\bar{\mathcal{K}}}$, then $Du_{\bar{\mathcal{K}}} \in \operatorname{Im} D$ can be decomposed as $Du_{\bar{\mathcal{K}}} = a + b$ for $a \in A$ and $b \in B$. These two components can be mapped to $\bar{\mathcal{K}}$ via the isomorphism above to some $\alpha$ and $\beta$ such that $D|_{\bar{\mathcal{K}}}\alpha = a$ and $D|_{\bar{\mathcal{K}}}\beta = b$. So $\alpha \in D^{-1}(A) \cap \bar{K}$ and $\beta \in D^{-1}(B) \cap \bar{K}$, and $D|_{\bar{\mathcal{K}}} u_{\bar{\mathcal{K}}} = Du_{\bar{\mathcal{K}}} = a + b = D|_{\bar{\mathcal{K}}}\alpha + D|_{\bar{\mathcal{K}}}\beta = D|_{\bar{\mathcal{K}}}(\alpha + \beta)$, so $u_{\bar{\mathcal{K}}} = \alpha + \beta$ since $D|_{\bar{\mathcal{K}}}$ is an isomorphism. to check

In the text, exact $= A$ and harmonic $= B$. □

## References


[Can+11] Ozan Candogan et al. "Flows and Decompositions of Games: Harmonic and Potential Games". In: *Mathematics of Operations Research* 36.3 (Aug. 2011), pp. 474–503. DOI: 10.1287/moor.1110.0500 (cit. on pp. 1, 2, 6, 15, 18).

[DLM05] Mathieu Desbrun, Melvin Leok, and Jerrold E. Marsden. "Discrete Poincaré Lemma". In: *Applied Numerical Mathematics*. Tenth Seminar on Numerical Solution of Differential and Differntial-Algebraic Euqations (NUMDIFF-10) 53.2 (May 2005), pp. 231–248. ISSN: 0168-9274. DOI: 10.1016/j.apnum.2004.09.035. (Visited on 01/29/2023) (cit. on p. 16).

[Eck44] Beno Eckmann. "Harmonische Funktionen und Randwertaufgaben in einem Komplex". In: *Commentarii Mathematici Helvetici* 17.1 (Dec. 1944), pp. 240–255. ISSN: 1420-8946. DOI: 10.1007/BF02566245. (Visited on 09/12/2023) (cit. on p. 1).

[Fri96] Joel Friedman. "Computing Betti Numbers via Combinatorial Laplacians". In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing - STOC '96*. Philadelphia, Pennsylvania, United States: ACM Press, 1996, pp. 386–391. ISBN: 978-0-89791-785-8. DOI: 10.1145/237814.237985. (Visited on 09/12/2023) (cit. on p. 1).

[Jia+11] Xiaoye Jiang et al. "Statistical Ranking and Combinatorial Hodge Theory". In: *Mathematical Programming* 127.1 (2011), pp. 203–244 (cit. on pp. 1, 15).

[KS05] K. Kamaraj and K. C. Sivakumar. "Moore-Penrose Inverse in an Indefinite Inner Product Space". In: *Journal of Applied Mathematics and Computing* 19.1 (Mar. 2005), pp. 297–310. ISSN: 1865-2085. DOI: 10.1007/BF02935806. (Visited on 02/13/2023) (cit. on p. 19).

[Mun84] James R Munkres. *Elements of Algebraic Topology*. Perseus Books, 1984 (cit. on pp. 1–3).

[RAG08] Steven Roman, S Axler, and FW Gehring. *Advanced Linear Algebra*. 3rd ed. Springer, 2008 (cit. on pp. 19, 21).