Spanning trees and the critical group of simplicial complexes

Art Duval[1] Caroline Klivans[2] Jeremy Martin[3]

[1]University of Texas at El Paso

[2]University of Chicago

[3]University of Kansas

Discrete CATS seminar
University of Kentucky
March 30, 2011

Counting weighted spanning trees of K_n

Theorem [Cayley]: K_n has n^{n-2} spanning trees.

T spanning tree: set of edges containing all vertices and

1. connected ($\tilde{H}_0(T) = 0$)
2. no cycles ($\tilde{H}_1(T) = 0$)
3. $|T| = n - 1$

Note: Any two conditions imply the third.

Counting weighted spanning trees of K_n

Theorem [Cayley]: K_n has n^{n-2} spanning trees.

T spanning tree: set of edges containing all vertices and

1. connected ($\tilde{H}_0(T) = 0$)
2. no cycles ($\tilde{H}_1(T) = 0$)
3. $|T| = n - 1$

Note: Any two conditions imply the third.

Weighting

vertices? Silly ($n^{n-2}(x_1 \cdots x_n)$)

Counting weighted spanning trees of K_n

Theorem [Cayley]: K_n has n^{n-2} spanning trees.

T spanning tree: set of edges containing all vertices and

1. connected ($\tilde{H}_0(T) = 0$)
2. no cycles ($\tilde{H}_1(T) = 0$)
3. $|T| = n - 1$

Note: Any two conditions imply the third.

Weighting

vertices? Silly ($n^{n-2}(x_1 \cdots x_n)$)

edges? No nice structure (can't see n^{n-2})

Counting weighted spanning trees of K_n

Theorem [Cayley]: K_n has n^{n-2} spanning trees.

T spanning tree: set of edges containing all vertices and

1. connected ($\tilde{H}_0(T) = 0$)
2. no cycles ($\tilde{H}_1(T) = 0$)
3. $|T| = n - 1$

Note: Any two conditions imply the third.

Weighting

vertices? Silly ($n^{n-2}(x_1 \cdots x_n)$)

edges? No nice structure (can't see n^{n-2})

both! $\operatorname{wt} T = \prod_{e \in T} \operatorname{wt} e = \prod_{e \in T} (\prod_{v \in e} x_v)$ Prüfer coding

Counting weighted spanning trees of K_n

Theorem [Cayley]: K_n has n^{n-2} spanning trees.

T spanning tree: set of edges containing all vertices and

1. connected ($\tilde{H}_0(T) = 0$)
2. no cycles ($\tilde{H}_1(T) = 0$)
3. $|T| = n - 1$

Note: Any two conditions imply the third.

Weighting

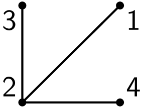
vertices? Silly ($n^{n-2}(x_1 \cdots x_n)$)

edges? No nice structure (can't see n^{n-2})

both! wt $T = \prod_{e \in T} \text{wt}\, e = \prod_{e \in T}(\prod_{v \in e} x_v)$ Prüfer coding

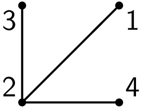
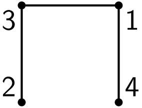
$$\sum_{T \in ST(K_n)} \text{wt}\, T = (x_1 \cdots x_n)(x_1 + \cdots + x_n)^{n-2}$$

Example: K_4

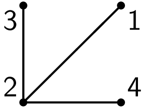
- 4 trees like: $T =$ 

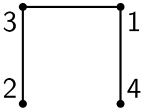
wt $T = (x_1 x_2 x_3 x_4) x_2^2$

Example: K_4

- 4 trees like: $T =$  wt $T = (x_1 x_2 x_3 x_4) x_2^2$
- 12 trees like: $T =$  wt $T = (x_1 x_2 x_3 x_4) x_1 x_3$

Example: K_4

- 4 trees like: $T =$ 

wt $T = (x_1x_2x_3x_4)x_2^2$
- 12 trees like: $T =$ 

wt $T = (x_1x_2x_3x_4)x_1x_3$

Total is $(x_1x_2x_3x_4)(x_1 + x_2 + x_3 + x_4)^2$.

Laplacian

Definition The Laplacian matrix of graph G, denoted by $L(G)$.

Laplacian

Definition The Laplacian matrix of graph G, denoted by $L\,(G)$.

Defn 1: $L(G) = D(G) - A(G)$

$$D(G) = \text{diag}(\deg v_1, \ldots, \deg v_n)$$
$$A(G) = \text{adjacency matrix}$$

Laplacian

Definition The Laplacian matrix of graph G, denoted by $L\ (G)$.

Defn 1: $L(G) = D(G) - A(G)$

$$D(G) = \mathrm{diag}(\deg v_1, \ldots, \deg v_n)$$

$$A(G) = \text{adjacency matrix}$$

Defn 2: $L(G) = \partial(G)\partial(G)^T$

$$\partial(G) = \text{incidence matrix (boundary matrix)}$$

Laplacian

Definition The reduced Laplacian matrix of graph G, denoted by $L_r(G)$.

Defn 1: $L(G) = D(G) - A(G)$

$$D(G) = \mathrm{diag}(\deg v_1, \ldots, \deg v_n)$$

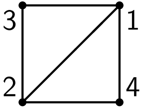
$$A(G) = \text{adjacency matrix}$$

Defn 2: $L(G) = \partial(G)\partial(G)^T$

$$\partial(G) = \text{incidence matrix (boundary matrix)}$$

"Reduced": remove rows/columns corresponding to any one vertex

Example



$$\partial = \begin{array}{c|ccccc} & 12 & 13 & 14 & 23 & 24 \\ \hline 1 & -1 & -1 & -1 & 0 & 0 \\ 2 & 1 & 0 & 0 & -1 & -1 \\ 3 & 0 & 1 & 0 & 1 & 0 \\ 4 & 0 & 0 & 1 & 0 & 1 \end{array}$$

$$L = \begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ -1 & -1 & 0 & 2 \end{pmatrix}$$

Matrix-Tree Theorems

Version I Let $0, \lambda_1, \lambda_2, \ldots, \lambda_{n-1}$ be the eigenvalues of L. Then G has

$$\frac{\lambda_1 \lambda_2 \cdots \lambda_{n-1}}{n}$$

spanning trees.

Version II G has $|\det L_r(G)|$ spanning trees

Proof [Version II]

$$\det L_r(G) = \det \partial_r(G)\partial_r(G)^T = \sum_T (\det \partial_r(T))^2$$

$$= \sum_T (\pm 1)^2$$

by Binet-Cauchy

Example: K_n

$$L(K_n) = nI - J \qquad (n \times n);$$
$$L_r(K_n) = nI - J \qquad (n-1 \times n-1)$$

Example: K_n

$$L(K_n) = nI - J \qquad (n \times n);$$
$$L_r(K_n) = nI - J \qquad (n-1 \times n-1)$$

Version I: Eigenvalues of L are $n - n$ (multiplicity 1), $n - 0$ (multiplicity $n - 1$), so

$$\frac{n^{n-1}}{n} = n^{n-2}$$

Example: K_n

$$L(K_n) = nI - J \qquad (n \times n);$$
$$L_r(K_n) = nI - J \qquad (n-1 \times n-1)$$

Version I: Eigenvalues of L are $n - n$ (multiplicity 1), $n - 0$ (multiplicity $n-1$), so

$$\frac{n^{n-1}}{n} = n^{n-2}$$

Version II:

$$\begin{aligned} \det L_r &= \prod \text{eigenvalues} \\ &= (n-0)^{(n-1)-1}(n-(n-1)) \\ &= n^{n-2} \end{aligned}$$

Weighted Matrix-Tree Theorem

$$\sum_{T \in ST(G)} \mathrm{wt}\, T = |\det \hat{L}_r(G)|,$$

where $\hat{L}$ is weighted Laplacian.

Defn 1: $\hat{L}(G) = \hat{D}(G) - \hat{A}(G)$

$\hat{D}(G) = \mathrm{diag}(\hat{\deg} v_1, \ldots, \hat{\deg} v_n)$

$\hat{\deg} v_i = \sum_{v_i v_j \in E} x_i x_j$

$\hat{A}(G) =$ adjacency matrix

(entry $x_i x_j$ for edge $v_i v_j$)

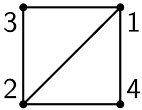
Defn 2: $\hat{L}(G) = \partial(G) B(G) \partial(G)^T$

$\partial(G) =$ incidence matrix

$B(G)$ diagonal, indexed by edges,

entry $\pm x_i x_j$ for edge $v_i v_j$

Example



$$\hat{L} = \begin{pmatrix} 1(2+3+4) & -12 & -13 & -14 \\ -12 & 2(1+3+4) & -23 & -24 \\ -13 & -23 & 3(1+2) & 0 \\ -14 & -24 & 0 & 4(1+2) \end{pmatrix}$$

$$\det \hat{L}_r = (1234)(1+2)(1+2+3+4)$$

Complete skeleta of simplicial complexes

Simplicial complex $\Delta \subseteq 2^V$;

$$F \subseteq G \in \Delta \Rightarrow F \in \Delta.$$

Complete skeleta of simplicial complexes

Simplicial complex $\Delta \subseteq 2^V$;
$F \subseteq G \in \Delta \Rightarrow F \in \Delta$.

Complete skeleton The d-dimensional complete complex on n vertices, *i.e.*,

$$K_n^d = \{F \subseteq V \colon |F| \leq d+1\}$$

(so $K_n = K_n^1$).

Simplicial spanning trees of K_n^d [Kalai, '83]

$\Upsilon \subseteq K_n^d$ is a **simplicial spanning tree** of K_n^d when:

0. $\Upsilon_{(d-1)} = K_n^{d-1}$ ("spanning");
1. $\tilde{H}_{d-1}(\Upsilon; \mathbb{Z})$ is a finite group ("connected");
2. $\tilde{H}_d(\Upsilon; \mathbb{Z}) = 0$ ("acyclic");
3. $|\Upsilon| = \binom{n-1}{d}$ ("count").

- If 0. holds, then any two of 1., 2., 3. together imply the third condition.
- When $d = 1$, coincides with usual definition.

Counting simplicial spanning trees of K_n^d

Conjecture [Bolker '76]

$$\sum_{\Upsilon \in SST(K_n^d)} \qquad = n^{\binom{n-2}{d}}$$

Counting simplicial spanning trees of K_n^d

Theorem [Kalai '83]

$$\sum_{\Upsilon \in SST(K_n^d)} |\tilde{H}_{d-1}(\Upsilon)|^2 = n^{\binom{n-2}{d}}$$

Weighted simplicial spanning trees of K_n^d

As before,

$$\operatorname{wt}\Upsilon = \prod_{F \in \Upsilon} \operatorname{wt} F = \prod_{F \in \Upsilon} \Big(\prod_{v \in F} x_v \Big)$$

Example:

$$\Upsilon = \{123, 124, 125, 134, 135, 245\}$$

$$\operatorname{wt}\Upsilon = x_1^5 x_2^4 x_3^3 x_4^3 x_5^3$$

Weighted simplicial spanning trees of K_n^d

As before,

$$\mathrm{wt}\,\Upsilon = \prod_{F\in\Upsilon} \mathrm{wt}\,F = \prod_{F\in\Upsilon}\Big(\prod_{v\in F} x_v\Big)$$

Example:

$$\Upsilon = \{123, 124, 125, 134, 135, 245\}$$

$$\mathrm{wt}\,\Upsilon = x_1^5 x_2^4 x_3^3 x_4^3 x_5^3$$

Theorem [Kalai, '83]

$$\sum_{T\in SST(K_n^d)} |\tilde{H}_{d-1}(T)|^2 (\mathrm{wt}\,T) = (x_1\cdots x_n)^{\binom{n-2}{d-1}} (x_1+\cdots+x_n)^{\binom{n-2}{d}}$$

Weighted simplicial spanning trees of K_n^d

As before,

$$\operatorname{wt}\Upsilon = \prod_{F\in\Upsilon} \operatorname{wt} F = \prod_{F\in\Upsilon} \Big(\prod_{v\in F} x_v\Big)$$

Example:

$$\Upsilon = \{123, 124, 125, 134, 135, 245\}$$

$$\operatorname{wt}\Upsilon = x_1^5 x_2^4 x_3^3 x_4^3 x_5^3$$

Theorem [Kalai, '83]

$$\sum_{T\in SST(K_n^d)} |\tilde{H}_{d-1}(T)|^2 (\operatorname{wt} T) = (x_1\cdots x_n)^{\binom{n-2}{d-1}} (x_1+\cdots+x_n)^{\binom{n-2}{d}}$$

(Adin ('92) did something similar for complete r-partite complexes.)

Proof

Proof uses determinant of reduced Laplacian of K_n^d. "Reduced" now means pick one vertex, and then remove rows/columns corresponding to all $(d-1)$-dimensional faces containing that vertex.

$L = \partial\partial^T$

$\partial\colon \Delta_d \to \Delta_{d-1}$ boundary

$\partial^T: \Delta_{d-1} \to \Delta_d$ coboundary

Example $n = 4, d = 2$

$$\partial^T = \begin{array}{c|cccccc} & 12 & 13 & 14 & 23 & 24 & 34 \\ \hline 123 & -1 & 1 & 0 & -1 & 0 & 0 \\ 124 & -1 & 0 & 1 & 0 & -1 & 0 \\ 134 & 0 & -1 & 1 & 0 & 0 & -1 \\ 234 & 0 & 0 & 0 & -1 & 1 & -1 \end{array}$$

$$L = \begin{pmatrix} 2 & -1 & -1 & 1 & 1 & 0 \\ -1 & 2 & -1 & -1 & 0 & 1 \\ -1 & -1 & 2 & 0 & -1 & -1 \\ 1 & -1 & 0 & 2 & -1 & 1 \\ 1 & 0 & -1 & -1 & 2 & -1 \\ 0 & 1 & -1 & 1 & -1 & 2 \end{pmatrix}$$

Simplicial spanning trees of arbitrary simplicial complexes

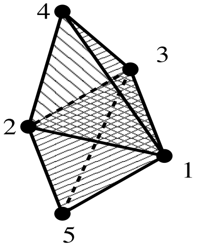
Let Δ be a d-dimensional simplicial complex.
$\Upsilon \subseteq \Delta$ is a **simplicial spanning tree** of Δ when:

0. $\Upsilon_{(d-1)} = \Delta_{(d-1)}$ ("spanning");
1. $\tilde{H}_{d-1}(\Upsilon; \mathbb{Z})$ is a finite group ("connected");
2. $\tilde{H}_d(\Upsilon; \mathbb{Z}) = 0$ ("acyclic");
3. $f_d(\Upsilon) = f_d(\Delta) - \tilde{\beta}_d(\Delta) + \tilde{\beta}_{d-1}(\Delta)$ ("count").

- If 0. holds, then any two of 1., 2., 3. together imply the third condition.
- When $d = 1$, coincides with usual definition.

Example

Bipyramid with equator, $\langle 123, 124, 125, 134, 135, 234, 235 \rangle$



Let's figure out all its simplicial spanning trees.

Acyclic in Positive Codimension (APC)

- Denote by $SST(\Delta)$ the set of simplicial spanning trees of Δ.
- **Proposition** $SST(\Delta) \neq \emptyset$ iff Δ is **APC**, *i.e.* (equivalently)
  - homology type of wedge of spheres;
  - $\tilde{H}_j(\Delta; \mathbb{Z})$ is finite for all $j < \dim \Delta$.
- Many interesting complexes are APC.

Simplicial Matrix-Tree Theorem — Version I

- Δ a d-dimensional APC simplicial complex
- $(d-1)$-dimensional **(up-down) Laplacian** $L_{d-1} = \partial_{d-1}\partial_{d-1}^T$
- s_d = product of nonzero eigenvalues of L_{d-1}.

Theorem [DKM]

$$h_d := \sum_{\Upsilon \in SST(\Delta)} |\tilde{H}_{d-1}(\Upsilon)|^2 = \frac{s_d}{h_{d-1}} |\tilde{H}_{d-2}(\Delta)|^2$$

Simplicial Matrix-Tree Theorem — Version II

- $\Gamma \in SST(\Delta_{(d-1)})$
- ∂_Γ = restriction of ∂_d to faces not in Γ
- reduced Laplacian $L_\Gamma = \partial_\Gamma \partial^T{}_\Gamma$

Theorem [DKM]

$$h_d \ = \sum_{\Upsilon \in SST(\Delta)} |\tilde{H}_{d-1}(\Upsilon)|^2 \ = \ \frac{|\tilde{H}_{d-2}(\Delta;\mathbb{Z})|^2}{|\tilde{H}_{d-2}(\Gamma;\mathbb{Z})|^2} \det L_\Gamma .$$

Note: The $|\tilde{H}_{d-2}|$ terms are often trivial.

Bipyramid again

$\Gamma = 12, 13, 14, 15$ spanning tree of 1-skeleton

Bipyramid again

$\Gamma = 12, 13, 14, 15$ spanning tree of 1-skeleton

$$L_\Gamma = \begin{array}{c|ccccc} & 23 & 24 & 25 & 34 & 35 \\ \hline 23 & 3 & -1 & -1 & 1 & 1 \\ 24 & -1 & 2 & 0 & -1 & 0 \\ 25 & -1 & 0 & 2 & 0 & -1 \\ 34 & 1 & -1 & 0 & 2 & 0 \\ 35 & 1 & 0 & -1 & 0 & 2 \end{array}$$

Bipyramid again

$\Gamma = 12, 13, 14, 15$ spanning tree of 1-skeleton

$$L_\Gamma = \begin{array}{c|ccccc} & 23 & 24 & 25 & 34 & 35 \\ \hline 23 & 3 & -1 & -1 & 1 & 1 \\ 24 & -1 & 2 & 0 & -1 & 0 \\ 25 & -1 & 0 & 2 & 0 & -1 \\ 34 & 1 & -1 & 0 & 2 & 0 \\ 35 & 1 & 0 & -1 & 0 & 2 \end{array}$$

$\det L_\Gamma = 15.$

Weighted Simplicial Matrix-Tree Theorems

- Introduce an indeterminate x_F for each face $F \in \Delta$
- Weighted boundary $\hat{\partial}$: multiply column F of (usual) ∂ by x_F
- $\hat{\partial}_\Gamma$ = restriction of $\hat{\partial}_d$ to faces not in Γ
- Weighted reduced Laplacian $\hat{L}_\Gamma = \hat{\partial}_\Gamma \hat{\partial^T}_\Gamma$

Theorem [DKM]

$$\hat{h}_d := \sum_{\Upsilon \in SST(\Delta)} |\tilde{H}_{d-1}(\Upsilon)|^2 \prod_{F \in \Upsilon} x_F^2 = \frac{\hat{s}_d}{\hat{h}_{d-1}} |\tilde{H}_{d-2}(\Delta)|^2$$

$$\hat{h}_d = \frac{|\tilde{H}_{d-2}(\Delta;\mathbb{Z})|^2}{|\tilde{H}_{d-2}(\Gamma;\mathbb{Z})|^2} \det \hat{L}_\Gamma.$$

Bipyramid, with weights

$L_\Gamma =$

	23	24	25	34	35
23	23(1+4+5)	-234	-235	234	235
24	-234	24(1+3)	0	-234	0
25	-235	0	25(1+3)	0	-235
34	234	234	0	(1+2)34	0
35	235	0	-235	0	(1+2)35

Bipyramid, with weights

$L_\Gamma =$

	23	24	25	34	35
23	23(1+4+5)	-234	-235	234	235
24	-234	24(1+3)	0	-234	0
25	-235	0	25(1+3)	0	-235
34	234	234	0	(1+2)34	0
35	235	0	-235	0	(1+2)35

$$\det L_\Gamma = (23)(24)(25)(34)(35) \times (1)^3(1+2+3)(1+2+3+4+5).$$

Nice families of complexes

Shifted complexes Bipyramid is an example. Their weighted spanning tree enumerator has a nice (if somewhat involved) factorization [DKM].

Nice families of complexes

Shifted complexes Bipyramid is an example. Their weighted spanning tree enumerator has a nice (if somewhat involved) factorization [DKM].

Matroid complexes They also appear to have a nice factorization, but no specific conjectures.

Nice families of complexes

Shifted complexes Bipyramid is an example. Their weighted spanning tree enumerator has a nice (if somewhat involved) factorization [DKM].

Matroid complexes They also appear to have a nice factorization, but no specific conjectures.

Color-shifted complexes They also appear to have a nice factorization, but no specific conjectures.

Nice families of complexes

Shifted complexes Bipyramid is an example. Their weighted spanning tree enumerator has a nice (if somewhat involved) factorization [DKM].

Matroid complexes They also appear to have a nice factorization, but no specific conjectures.

Color-shifted complexes They also appear to have a nice factorization, but no specific conjectures.

Cubical complexes We can generalize the definition pretty easily to cellular complexes, and then shifted cubical complexes also have a nice factorization.

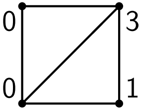
Sandpiles and chip-firing

Motivation Think of a sandpile, with grains of sand on vertices of a graph. When the pile at one place is too large, it topples, sending grains to all its neighbors.

Sandpiles and chip-firing

Motivation Think of a sandpile, with grains of sand on vertices of a graph. When the pile at one place is too large, it topples, sending grains to all its neighbors.

Abstraction Graph G with vertices $v_1, \ldots, v_n$. Degree of v_i is d_i. Place $c_i \in \mathbb{Z}$ chips (grains of sand) on v_i.



Sandpiles and chip-firing

Motivation Think of a sandpile, with grains of sand on vertices of a graph. When the pile at one place is too large, it topples, sending grains to all its neighbors.

Abstraction Graph G with vertices $v_1, \ldots, v_n$. Degree of v_i is d_i. Place $c_i \in \mathbb{Z}$ chips (grains of sand) on v_i.

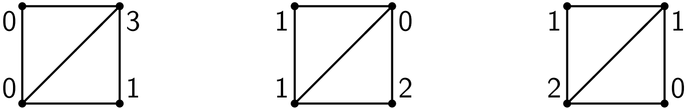
Toppling If $c_i \geq d_i$, then v_i may fire by sending one chip to each of its neighbors.

Sandpiles and chip-firing

Motivation Think of a sandpile, with grains of sand on vertices of a graph. When the pile at one place is too large, it topples, sending grains to all its neighbors.

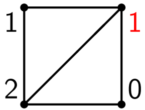
Abstraction Graph G with vertices $v_1, \ldots, v_n$. Degree of v_i is d_i. Place $c_i \in \mathbb{Z}$ chips (grains of sand) on v_i.

Toppling If $c_i \geq d_i$, then v_i may fire by sending one chip to each of its neighbors.



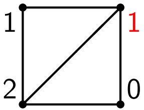
Source vertex

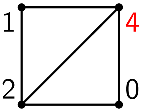
- To keep things going, pick one vertex v_r to be a source vertex. We can always add chips to v_r.



Source vertex

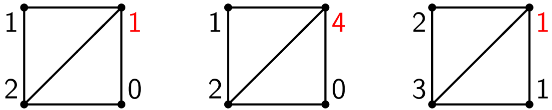
- To keep things going, pick one vertex v_r to be a source vertex. We can always add chips to v_r.
- Put another way: c_r can be any value.





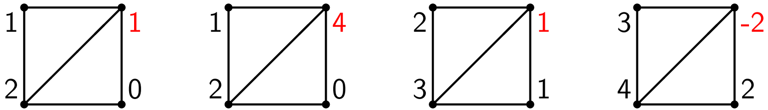
Source vertex

- To keep things going, pick one vertex v_r to be a source vertex. We can always add chips to v_r.
- Put another way: c_r can be any value.
- We might think $c_r \leq 0$, and $c_i \geq 0$ when $i \neq r$, or that v_r can fire even when $c_r \leq d_r$.



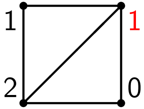
Source vertex

- To keep things going, pick one vertex v_r to be a source vertex. We can always add chips to v_r.
- Put another way: c_r can be any value.
- We might think $c_r \leq 0$, and $c_i \geq 0$ when $i \neq r$, or that v_r can fire even when $c_r \leq d_r$.



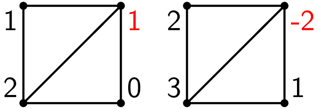
Critical configurations

- A configuration is stable when no vertex (except the source vertex) can fire.



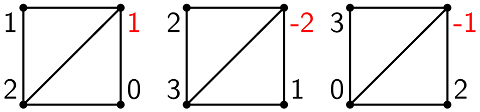
Critical configurations

- A configuration is stable when no vertex (except the source vertex) can fire.
- A configuration is recurrent when a series of topplings leads back to that configuration, without letting any vertex (except the source vertex) go negative.



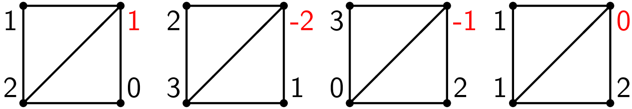
Critical configurations

- A configuration is stable when no vertex (except the source vertex) can fire.
- A configuration is recurrent when a series of topplings leads back to that configuration, without letting any vertex (except the source vertex) go negative.
- A configuration is critical when it is stable and recurrent.



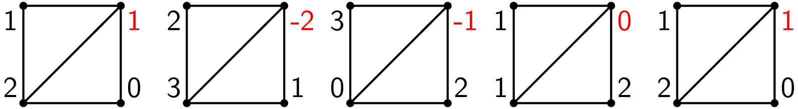
Critical configurations

- A configuration is stable when no vertex (except the source vertex) can fire.
- A configuration is recurrent when a series of topplings leads back to that configuration, without letting any vertex (except the source vertex) go negative.
- A configuration is critical when it is stable and recurrent.



Critical configurations

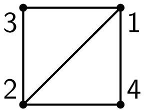
- A configuration is stable when no vertex (except the source vertex) can fire.
- A configuration is recurrent when a series of topplings leads back to that configuration, without letting any vertex (except the source vertex) go negative.
- A configuration is critical when it is stable and recurrent.



Fact: Every configuration topples to a unique critical configuration.

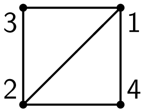
Laplacian

Let's make a matrix of how chips move when each vertex fires:



Laplacian

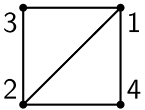
Let's make a matrix of how chips move when each vertex fires:



$$\begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ -1 & -1 & 0 & 2 \end{pmatrix} = D - A,$$

Laplacian

Let's make a matrix of how chips move when each vertex fires:



$$\begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ -1 & -1 & 0 & 2 \end{pmatrix} = D - A,$$

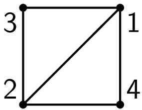
which is the Laplacian matrix

$$L = D - A = \partial\partial^T$$

where ∂ is the boundary (or incidence) matrix.

Laplacian

Let's make a matrix of how chips move when each vertex fires:



$$\begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ -1 & -1 & 0 & 2 \end{pmatrix} = D - A,$$

which is the Laplacian matrix

$$L = D - A = \partial\partial^T$$

where ∂ is the boundary (or incidence) matrix.
So firing v is subtracting Lv (row/column v from L) from $(c_1, \ldots, c_n)$.

Kernel ∂

- Did you notice?: Sum of chips stays constant.

Kernel ∂

- Did you notice?: Sum of chips stays constant.
- Also recall value of the source vertex can be anything, including negative (other vertices should stay positive).

Kernel ∂

- Did you notice?: Sum of chips stays constant.
- Also recall value of the source vertex can be anything, including negative (other vertices should stay positive).
- So we may as well insist that

$$\sum_i c_i = 0.$$

In other words, $\partial c = 0$, i.e., $c \in \ker \partial$.

Kernel ∂

- Did you notice?: Sum of chips stays constant.
- Also recall value of the source vertex can be anything, including negative (other vertices should stay positive).
- So we may as well insist that

$$\sum_i c_i = 0.$$

  In other words, $\partial c = 0$, i.e., $c \in \ker \partial$.
- We can pick $c_i, i \neq r$, arbitrarily, and keep $c \in \ker \partial$ by picking c_r appropriately.

Kernel ∂

- Did you notice?: Sum of chips stays constant.
- Also recall value of the source vertex can be anything, including negative (other vertices should stay positive).
- So we may as well insist that

$$\sum_i c_i = 0.$$

  In other words, $\partial c = 0$, i.e., $c \in \ker \partial$.
- We can pick $c_i, i \neq r$, arbitrarily, and keep $c \in \ker \partial$ by picking c_r appropriately.

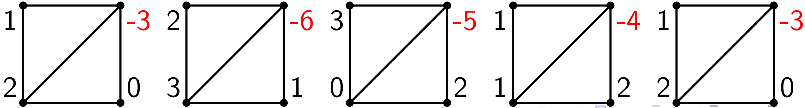
Kernel ∂

- Did you notice?: Sum of chips stays constant.
- Also recall value of the source vertex can be anything, including negative (other vertices should stay positive).
- So we may as well insist that

$$\sum_i c_i = 0.$$

In other words, $\partial c = 0$, i.e., $c \in \ker \partial$.

- We can pick $c_i, i \neq r$, arbitrarily, and keep $c \in \ker \partial$ by picking c_r appropriately.



Critical group

- Consider two configurations (in $\ker \partial$) to be equivalent when you can get from one to the other by chip-firing.

Critical group

- Consider two configurations (in $\ker \partial$) to be equivalent when you can get from one to the other by chip-firing.
- Recall every configuration is equivalent to a critical configuration.

Critical group

- Consider two configurations (in $\ker \partial$) to be equivalent when you can get from one to the other by chip-firing.
- Recall every configuration is equivalent to a critical configuration.
- This equivalence means adding/subtracting integer multiples of Lv_i.

Critical group

- Consider two configurations (in $\ker \partial$) to be equivalent when you can get from one to the other by chip-firing.
- Recall every configuration is equivalent to a critical configuration.
- This equivalence means adding/subtracting integer multiples of Lv_i.
- In other words, instead of $\ker \partial$, we look at

$$K(G) := \ker \partial / \operatorname{im} L$$

the critical group. (It is a graph invariant.)

Reduced Laplacian and spanning trees

Theorem (Biggs '99)

$$K := (\ker \partial)/(\operatorname{im} L) \cong \mathbb{Z}^{n-1}/L_r,$$

where L_r denotes reduced Laplacian; remove row and column corresponding to source vertex.

Reduced Laplacian and spanning trees

Theorem (Biggs '99)

$$K := (\ker \partial)/(\operatorname{im} L) \cong \mathbb{Z}^{n-1}/L_r,$$

where L_r denotes reduced Laplacian; remove row and column corresponding to source vertex.

Corollary

$|K(G)|$ is the number of spanning trees of G.

Reduced Laplacian and spanning trees

Theorem (Biggs '99)

$$K := (\ker \partial)/(\operatorname{im} L) \cong \mathbb{Z}^{n-1}/L_r,$$

where L_r denotes reduced Laplacian; remove row and column corresponding to source vertex.

Corollary

$|K(G)|$ is the number of spanning trees of G.

Proof.

If M is a full rank t-dimensional matrix, then

$$|(\mathbb{Z}^t)/(\operatorname{im} M)| = \pm \det M$$

Reduced Laplacian and spanning trees

Theorem (Biggs '99)

$$K := (\ker \partial)/(\operatorname{im} L) \cong \mathbb{Z}^{n-1}/L_r,$$

where L_r denotes reduced Laplacian; remove row and column corresponding to source vertex.

Corollary

$|K(G)|$ is the number of spanning trees of G.

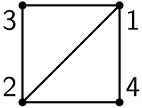
Proof.

If M is a full rank t-dimensional matrix, then

$$|(\mathbb{Z}^t)/(\operatorname{im} M)| = \pm \det M$$

and $|\det L_r|$ counts spanning trees. □

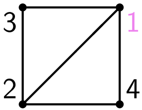
Example



$$\partial = \begin{array}{c|ccccc} & 12 & 13 & 14 & 23 & 24 \\ \hline 1 & -1 & -1 & -1 & 0 & 0 \\ 2 & 1 & 0 & 0 & -1 & -1 \\ 3 & 0 & 1 & 0 & 1 & 0 \\ 4 & 0 & 0 & 1 & 0 & 1 \end{array}$$

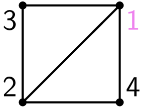
$$L = \begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ -1 & -1 & 0 & 2 \end{pmatrix}$$

Example



$$L = \begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ -1 & -1 & 0 & 2 \end{pmatrix}$$

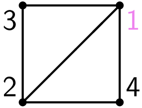
Example



$$L = \begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ -1 & -1 & 0 & 2 \end{pmatrix}$$

$$L_r = \begin{pmatrix} 3 & -1 & -1 \\ -1 & 2 & 0 \\ -1 & 0 & 2 \end{pmatrix}$$

Example



$$L = \begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ -1 & -1 & 0 & 2 \end{pmatrix}$$

$$L_r = \begin{pmatrix} 3 & -1 & -1 \\ -1 & 2 & 0 \\ -1 & 0 & 2 \end{pmatrix}$$

$\det L_r = 8$, and there are 8 spanning trees of this graph

Where have we seen this before?

Graphs

Where have we seen this before?

Graphs

- To count spanning trees, and compute critical group, use the determinant of the reduced Laplacian.

Where have we seen this before?

Graphs

- To count spanning trees, and compute critical group, use the determinant of the reduced Laplacian.
- Reduce Laplacian by removing a vertex.

Where have we seen this before?

Graphs

- To count spanning trees, and compute critical group, use the determinant of the reduced Laplacian.
- Reduce Laplacian by removing a vertex.

Simplicial complexes

Where have we seen this before?

Graphs

- To count spanning trees, and compute critical group, use the determinant of the reduced Laplacian.
- Reduce Laplacian by removing a vertex.

Simplicial complexes

- To count spanning trees, use the determinant of the reduced Laplacian.

Where have we seen this before?

Graphs

- To count spanning trees, and compute critical group, use the determinant of the reduced Laplacian.
- Reduce Laplacian by removing a vertex.

Simplicial complexes

- To count spanning trees, use the determinant of the reduced Laplacian.
- Reduce Laplacian by removing a $(d-1)$-dimensional spanning tree from up-down Laplacian.

Where have we seen this before?

Graphs

- To count spanning trees, and compute critical group, use the determinant of the reduced Laplacian.
- Reduce Laplacian by removing a vertex.

Simplicial complexes

- To count spanning trees, use the determinant of the reduced Laplacian.
- Reduce Laplacian by removing a $(d-1)$-dimensional spanning tree from up-down Laplacian.

Where have we seen this before?

Graphs

- To count spanning trees, and compute critical group, use the determinant of the reduced Laplacian.
- Reduce Laplacian by removing a vertex.

Simplicial complexes

- To count spanning trees, use the determinant of the reduced Laplacian.
- Reduce Laplacian by removing a $(d-1)$-dimensional spanning tree from up-down Laplacian.

So let's generalize critical groups to simplicial complexes, and see if they can be computed by reduced Laplacians.

Definition

Recall, for a graph G,

$$K(G) := \ker \partial / \operatorname{im} L.$$

Definition

Recall, for a graph G,

$$K(G) := \ker \partial / \operatorname{im} L.$$

Let Δ be a d-dimensional simplicial complex.

$$C_d(\Delta;\mathbb{Z}) \underset{\partial_d}{\overset{\partial_d^T}{\leftrightarrows}} C_{d-1}(\Delta;\mathbb{Z}) \xrightarrow{\partial_{d-1}} C_{d-2}(\Delta;\mathbb{Z}) \to \cdots$$

$$C_{d-1}(\Delta;\mathbb{Z}) \xrightarrow{L_{d-1}} C_{d-1}(\Delta;\mathbb{Z}) \xrightarrow{\partial_{d-1}} C_{d-2}(\Delta;\mathbb{Z}) \to \cdots$$

Definition

Recall, for a graph G,

$$K(G) := \ker \partial / \operatorname{im} L.$$

Let Δ be a d-dimensional simplicial complex.

$$C_d(\Delta;\mathbb{Z}) \underset{\partial_d}{\overset{\partial_d^T}{\leftrightarrows}} C_{d-1}(\Delta;\mathbb{Z}) \xrightarrow{\partial_{d-1}} C_{d-2}(\Delta;\mathbb{Z}) \to \cdots$$

$$C_{d-1}(\Delta;\mathbb{Z}) \xrightarrow{L_{d-1}} C_{d-1}(\Delta;\mathbb{Z}) \xrightarrow{\partial_{d-1}} C_{d-2}(\Delta;\mathbb{Z}) \to \cdots$$

Define

$$K(\Delta) := \ker \partial_{d-1} / \operatorname{im} L_{d-1},$$

where $L_{d-1} = \partial_d \partial_d^T$ is the $(d-1)$-dimensional up-down Laplacian.

Spanning trees

Theorem (DKM)

$$K(\Delta) := (\ker \partial_{d-1})/(\operatorname{im} L_{d-1}) \cong \mathbb{Z}^t / L_\Gamma$$

where Γ *is a torsion-free* $(d-1)$*-dimensional spanning tree,* L_Γ *is the reduced Laplacian (restriction to faces not in* Γ*), and* $t = \dim L_\Gamma$.

Spanning trees

Theorem (DKM)

$$K(\Delta) := (\ker \partial_{d-1})/(\operatorname{im} L_{d-1}) \cong \mathbb{Z}^t/L_\Gamma$$

where Γ *is a torsion-free* $(d-1)$*-dimensional spanning tree,* L_Γ *is the reduced Laplacian (restriction to faces not in* Γ*), and* $t = \dim L_\Gamma$*.*

Corollary

$|K(\Delta)|$ *is the torsion-weighted number of* d*-dimensional spanning trees of* Δ*.*

Proof.

$|K(\Delta)| = |(\mathbb{Z}^t)/L_\Gamma| = |\det L_\Gamma|$, which counts (torsion-weighted) spanning trees. $\square$

What does it look like?

$$K(\Delta) := \ker \partial_{d-1} / \operatorname{im} L_{d-1} \subseteq \mathbb{Z}^m$$

What does it look like?

$$K(\Delta) := \ker \partial_{d-1} / \operatorname{im} L_{d-1} \subseteq \mathbb{Z}^m$$

- Put integers on $(d-1)$-faces of Δ. Orient faces arbitrarily. $d = 2$: flow; $d = 3$: circulation; etc.

What does it look like?

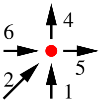
$$K(\Delta) := \ker \partial_{d-1} / \operatorname{im} L_{d-1} \subseteq \mathbb{Z}^m$$

- Put integers on $(d-1)$-faces of Δ. Orient faces arbitrarily. $d = 2$: flow; $d = 3$: circulation; etc.
- conservative flow

What does it look like?

$$K(\Delta) := \ker \partial_{d-1} / \operatorname{im} L_{d-1} \subseteq \mathbb{Z}^m$$

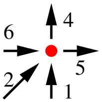
- Put integers on $(d-1)$-faces of Δ. Orient faces arbitrarily. $d = 2$: flow; $d = 3$: circulation; etc.
- conservative flow
  - $d = 2$: chips do not accumulate or deplete at any vertex;

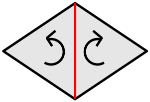


What does it look like?

$$K(\Delta) := \ker \partial_{d-1} / \operatorname{im} L_{d-1} \subseteq \mathbb{Z}^m$$

- Put integers on $(d-1)$-faces of Δ. Orient faces arbitrarily. $d=2$: flow; $d=3$: circulation; etc.
- conservative flow
  - $d=2$: chips do not accumulate or deplete at any vertex;
  - $d=3$: face circulation at each edge adds to zero.

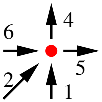


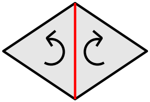


What does it look like?

$$K(\Delta) := \ker \partial_{d-1} / \operatorname{im} L_{d-1} \subseteq \mathbb{Z}^m$$

- Put integers on $(d-1)$-faces of Δ. Orient faces arbitrarily. $d = 2$: flow; $d = 3$: circulation; etc.
- conservative flow
  - $d = 2$: chips do not accumulate or deplete at any vertex;
  - $d = 3$: face circulation at each edge adds to zero.
- By theorem, just specify values off the spanning tree.

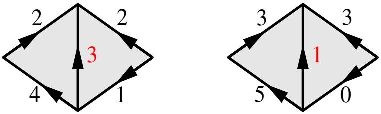




Firing faces

$$K(\Delta) := \ker \partial_{d-1} / \operatorname{im} L_{d-1} \subseteq \mathbb{Z}^m$$

Toppling/firing moves the flow to "neighboring" $(d-1)$-faces, across d-faces.



Open problem: Critical configurations?

- What are the critical configurations?

Open problem: Critical configurations?

- What are the critical configurations?
  - i.e., canonical set of representatives

Open problem: Critical configurations?

- What are the critical configurations?
  - i.e., canonical set of representatives
- We could pick any set of representatives; by definition, there is some sequence of firings taking any configuration to the representative.

Open problem: Critical configurations?

- What are the critical configurations?
  - i.e., canonical set of representatives
- We could pick any set of representatives; by definition, there is some sequence of firings taking any configuration to the representative.
- But this misses the sense of “critical”.

Open problem: Critical configurations?

- What are the critical configurations?
  - i.e., canonical set of representatives
- We could pick any set of representatives; by definition, there is some sequence of firings taking any configuration to the representative.
- But this misses the sense of “critical”.
- Main obstacle is idea of what is “positive”.

Example: Spheres

Theorem

If Δ is a sphere, with n facets, then $K(\Delta) \cong \mathbb{Z}_n$.

Example: Spheres

Theorem

If Δ is a sphere, with n facets, then $K(\Delta) \cong \mathbb{Z}_n$.

$K(\Delta) := \ker \partial_{d-1} / \operatorname{im} L_{d-1}$

Proof.

- $K(\Delta)$ is generated by boundaries of facets ∂F.

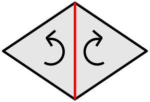
□

Example: Spheres

Theorem

If Δ is a sphere, with n facets, then $K(\Delta) \cong \mathbb{Z}_n$.

$K(\Delta) := \ker \partial_{d-1} / \operatorname{im} L_{d-1}$



Proof.

- $K(\Delta)$ is generated by boundaries of facets ∂F.
- In a sphere, the Laplacian of a ridge shows if facets F, G are adjacent, then $\partial F \equiv \pm \partial G \pmod{\operatorname{im} L}$.

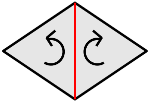
□

Example: Spheres

Theorem

If Δ is a sphere, with n facets, then $K(\Delta) \cong \mathbb{Z}_n$.

$K(\Delta) := \ker \partial_{d-1} / \operatorname{im} L_{d-1}$



Proof.

- $K(\Delta)$ is generated by boundaries of facets ∂F.
- In a sphere, the Laplacian of a ridge shows if facets F, G are adjacent, then $\partial F \equiv \pm \partial G \pmod{\operatorname{im} L}$.
- So $K(\Delta)$ has a single generator, so it is cyclic.

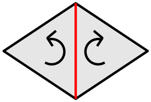
□

Example: Spheres

Theorem

If Δ is a sphere, with n facets, then $K(\Delta) \cong \mathbb{Z}_n$.

$K(\Delta) := \ker \partial_{d-1} / \operatorname{im} L_{d-1}$



Proof.

- $K(\Delta)$ is generated by boundaries of facets ∂F.
- In a sphere, the Laplacian of a ridge shows if facets F, G are adjacent, then $\partial F \equiv \pm \partial G \pmod{\operatorname{im} L}$.
- So $K(\Delta)$ has a single generator, so it is cyclic.
- $|K(\Delta)|$ is the number of spanning trees, and there is one tree for every facet (remove that facet for the tree)

$\square$

Final thought

Terry Pratchett, *The Colour of Magic*:
"Do you not know that what you belittle by the name *tree* is but the mere four-dimensional analogue of a whole multidimensional universe which—no, I can see you do not."

Final thought

Terry Pratchett, *The Colour of Magic*:
"Do you not know that what you belittle by the name *tree* is but the mere four-dimensional analogue of a whole multidimensional universe which—no, I can see you do not."

But, now, *you* do.