Spanning trees and the critical group of simplicial complexes

Art Duval\textsuperscript{1}  Caroline Klivans\textsuperscript{2}  Jeremy Martin\textsuperscript{3}

\textsuperscript{1}University of Texas at El Paso  
\textsuperscript{2}University of Chicago  
\textsuperscript{3}University of Kansas

Mathematics Seminar  
Reed College  
April 28, 2011
Spanning trees of $K_n$

Theorem (Cayley)

$K_n$ has $n^{n-2}$ spanning trees.
Spanning trees of $K_n$

Theorem (Cayley)

$K_n$ has $n^{n-2}$ spanning trees.

$T \subseteq E(K_n)$ is a spanning tree of $K_n$ when:

0. spanning: $T$ contains all vertices;
1. connected ($\tilde{H}_0(T) = 0$)
2. no cycles ($\tilde{H}_1(T) = 0$)
3. correct count: $|T| = n - 1$

If 0. holds, then any two of 1., 2., 3. together imply the third condition.
Example: $K_4$

- 4 trees like: $T = \{1, 2, 3, 4\}$
Example: $K_4$

- 4 trees like: $T =$

- 12 trees like: $T =$

Total is $16 = 4^2$. 

Duval, Klivans, Martin: Spanning trees and the critical group of simplicial complexes
Example: $K_4$

- 4 trees like: $T = \begin{align*}
    &3 \\
    2 &\quad 4 \\
    &3 \\
    1 &\quad 2 \\
\end{align*}$

- 12 trees like: $T = \begin{align*}
    &3 \\
    2 &\quad 4 \\
    &3 \\
    1 &\quad 2 \\
\end{align*}$

Total is $16 = 4^2$. 
Definition The Laplacian matrix of graph $G$, denoted by $L(G)$. \
\[ L(G) = D(G) - A(G) \]
\[ D(G) = \text{diag}(\deg v_1, \ldots, \deg v_n) \]
\[ A(G) = \text{adjacency matrix} \]
\[ \partial(G) \partial(G)^T = \text{incidence matrix (boundary matrix)} \]

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

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

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

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

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

Defn 1: $L(G) = D(G) - A(G)$  
$D(G) = \text{diag}(\deg v_1, \ldots, \deg v_n)$  
$A(G) = \text{adjacency matrix}$

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: \( L(G) = D(G) - A(G) \)

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

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

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

\[
\begin{align*}
L &= \begin{pmatrix}
3 & -1 & -1 & -1 \\
-1 & 3 & -1 & -1 \\
-1 & -1 & 2 & 0 \\
-1 & -1 & 0 & 2 \\
\end{pmatrix} \\
\partial &= \begin{pmatrix}
1 & -1 & -1 & -1 & 0 & 0 \\
2 & 1 & 0 & 0 & -1 & -1 \\
3 & 0 & 1 & 0 & 1 & 0 \\
4 & 0 & 0 & 1 & 0 & 1 \\
\end{pmatrix}
\end{align*}
\]
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: $K_n$

\[
\begin{align*}
L(K_n) &= nl - J & (n \times n); \\
L_r(K_n) &= nl - J & (n - 1 \times n - 1)
\end{align*}
\]
Example: \( K_n \)

\[
L(K_n) = nl - J \quad (n \times n);
\]

\[
L_r(K_n) = nl - J \quad (n - 1 \times n - 1)
\]

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: $K_n$

$$L(K_n) = nl - J \quad (n \times n);$$

$$L_r(K_n) = nl - J \quad (n - 1 \times n - 1)$$

**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:**

$$\det L_r = \prod \text{eigenvalues}$$

$$= (n - 0)^{(n-1)-1}(n - (n - 1))$$

$$= n^{n-2}$$
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.,
\[ K^d_n = \{ F \subseteq V : |F| \leq d + 1 \} \]
(so \( K_n = K^1_n \)).
Simplicial spanning trees of $K^d_n$ [Kalai, ’83]

$\Upsilon \subseteq K^d_n$ is a simplicial spanning tree of $K^d_n$ when:

0. $\Upsilon_{(d-1)} = K^{d-1}_n$ (“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]

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

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.
Counting simplicial spanning trees of $K_n^d$

**Theorem** [Kalai ’83]

$$\sum_{\gamma \in SST(K_n^d)} |\tilde{H}_{d-1}(\gamma)|^2 = n^{\binom{n-2}{d}}$$
Counting simplicial spanning trees of $K_n^d$

**Theorem** [Kalai ’83]

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

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.

$L = \partial \partial^T$

$\partial : \Delta_d \rightarrow \Delta_{d-1}$ boundary

$\partial^T : \Delta_{d-1} \rightarrow \Delta_d$ coboundary
Example $n = 4, d = 2$

\[ \partial^T = \begin{array}{cccccccc}
123 & 12 & 13 & 14 & 23 & 24 & 34 \\
123 & -1 & 1 & 0 & -1 & 0 & 0 \\
124 & -1 & 0 & 1 & 0 & -1 & 0 \\
134 & 0 & -1 & 1 & 0 & 0 & -1 \\
234 & 0 & 0 & 0 & -1 & 1 & -1 \\
\end{array} \]

\[ L = \begin{pmatrix}
2 & -1 & -1 & 1 & 1 & 0 \\
-1 & 2 & -1 & -1 & 0 & 1 \\
-1 & -1 & 2 & 0 & -1 & -1 \\
1 & -1 & 0 & 2 & -1 & 1 \\
1 & 0 & -1 & -1 & 2 & -1 \\
0 & 1 & -1 & 1 & -1 & 2 \\
\end{pmatrix} \]
Simplicial spanning trees of arbitrary simplicial complexes

Let $\Delta$ be a $d$-dimensional simplicial complex. $\Upsilon \subseteq \Delta$ is a simplicial spanning tree of $\Delta$ 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 $\Delta$.
- **Proposition** $SST(\Delta) \neq \emptyset$ iff $\Delta$ 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

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

**Theorem** [DKM '09]

$$h_d := \sum_{\gamma \in SST(\Delta)} |\tilde{H}_{d-1}(\gamma)|^2 = \frac{s_d}{h_{d-1}} |\tilde{H}_{d-2}(\Delta)|^2$$
Simplicial Matrix-Tree Theorem — Version II

- $\Gamma \in SST(\Delta_{d-1})$
- $\partial \Gamma$ = restriction of $\partial_d$ to faces not in $\Gamma$
- reduced Laplacian $L_\Gamma = \partial \Gamma \partial^T \Gamma$

**Theorem** [DKM '09]

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

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

\[ \Gamma = 12, 13, 14, 15 \] spanning tree of 1-skeleton
### Bipyramid again

\[ \Gamma = 12, 13, 14, 15 \text{ spanning tree of 1-skeleton} \]

\[
\begin{array}{c|cccccc}
 & 23 & 24 & 25 & 34 & 35 \\
\hline
23 & 3 & -1 & -1 & 1 & 1 \\
24 & -1 & 2 & 0 & -1 & 0 \\
25 & -1 & 0 & 2 & 0 & -1 \\
34 & 1 & -1 & 0 & 2 & 0 \\
35 & 1 & 0 & -1 & 0 & 2 \\
\end{array}
\]

\[ \text{det} L_\Gamma = 15. \]
Bipyramid again

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

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

\[ \det L_\Gamma = 15. \]
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.
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.

\[
\begin{array}{c}
\begin{array}{c}
0 \\
0
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
1 \\
1
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
3 \\
1
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
1 \\
2
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
0 \\
0
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
1 \\
2
\end{array}
\end{array}
\end{array}
\]
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.
To keep things going, pick one vertex $v_r$ to be a source vertex. We can always add chips to $v_r$. 

\[
\begin{array}{ccc}
1 & 1 \\
2 & 0
\end{array}
\]
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 \( c_r \leq d_r \).
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 $c_r \leq d_r$. 

1 1
2 0

1 4
2 0

2 1
3 1

3 -2
4 2
Critical configurations

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

\[
\begin{pmatrix}
0 & 1 & 1 \\
2 & 0 & 1 \\
1 & 1 & 0
\end{pmatrix}
\]

Fact: Every configuration topples to a unique critical configuration.
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.

![Graph with vertices and edge weights]

**Fact:** Every configuration topples to a unique critical configuration.

Duval, Klivans, Martin
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.

```
1 1 2
2 0 3
```
```
-2 3
1 0
```
```
-1
2
```
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.
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:

\[
\begin{pmatrix}
3 & 1 \\
2 & 4
\end{pmatrix}
\]

\[
\begin{pmatrix}
3 & -1 & -1 & -1 \\
-1 & 2 & -1 & -1 \\
-1 & -1 & 2 & 0 \\
-1 & -1 & 0 & 2
\end{pmatrix}
\]

which is the Laplacian matrix

\[
L = D - A = \partial T
\]

where \(\partial\) is the boundary (or incidence) matrix.
Let's make a matrix of how chips move when each vertex fires:

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

\[
D - A = \text{Laplacian matrix}
\]
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 $\partial$ is the boundary (or incidence) matrix.
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 \( \partial \) is the boundary (or incidence) matrix. So firing \( v \) is subtracting \( Lv \) (row/column \( v \) from \( L \)) from \((c_1, \ldots, c_n)\).
Kernel $\partial$

- Did you notice?: Sum of chips stays constant.
Kernel $\partial$

- 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 $\partial$

- 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 $\partial$

- 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 $\partial$

- 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 $\partial$

- 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 $Lv_i$. 

\[\text{Critical group} \quad \text{of simplicial complexes}\]
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

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

the critical group. (It is a graph invariant.)
Reduced Laplacian and spanning trees

Theorem (Biggs ’99)

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

where \( L_r \) denotes reduced Laplacian; remove row and column corresponding to source vertex.
Reduced Laplacian and spanning trees

Theorem (Biggs ’99)

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

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)

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

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

Corollary

\(|K(G)| \text{ is the number of spanning trees of } G.\)

Proof.

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

\[ |(\mathbb{Z}^t)/(\text{im} \ M)| = \pm \det M \]
Reduced Laplacian and spanning trees

Theorem (Biggs ’99)

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

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

Corollary

\(|K(G)| \text{ is the number of spanning trees of } G.\)

Proof.

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

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

and \( |\det L_r| \) counts spanning trees.
Example

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

\[ \partial = \begin{array}{c|cccccc} & 12 & 13 & 14 & 23 & 24 \\ \hline 1 & -1 & -1 & -1 & 0 & 0 \\ 2 & 1 & 0 & 0 & -1 & -1 \\ 3 & 0 & 1 & 0 & 1 & 0 \\ 4 & 0 & 0 & 1 & 0 & 1 \end{array} \]
Example

\[ \begin{array}{cccc}
3 & -1 & -1 & -1 \\
-1 & 3 & -1 & -1 \\
-1 & -1 & 2 & 0 \\
-1 & -1 & 0 & 2 \\
\end{array} \]

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

\[ \det L_r = 8, \text{ and there are 8 spanning trees of this graph} \]
Example

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

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

\[\det(L_r) = 8, \text{ and there are 8 spanning trees of this graph}\]
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

Duval, Klivans, Martin
Spanning trees and the critical group of 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 $G$,

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

Recall, for a graph $G$,

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

Let $\Delta$ be a $d$-dimensional simplicial complex.

$$C_d(\Delta; \mathbb{Z}) \overset{\partial_d^T}{\leftrightarrow} C_{d-1}(\Delta; \mathbb{Z}) \overset{\partial_{d-1}}{\rightarrow} C_{d-2}(\Delta; \mathbb{Z}) \rightarrow \cdots$$  

$$C_{d-1}(\Delta; \mathbb{Z}) \overset{L_{d-1}}{\rightarrow} C_{d-1}(\Delta; \mathbb{Z}) \overset{\partial_{d-1}}{\rightarrow} C_{d-2}(\Delta; \mathbb{Z}) \rightarrow \cdots$$
Definition

Recall, for a graph $G$,

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

Let $\Delta$ be a $d$-dimensional simplicial complex.

$$C_d(\Delta; \mathbb{Z}) \xleftrightarrow{\partial_d^T} C_{d-1}(\Delta; \mathbb{Z}) \xrightarrow{\partial_{d-1}} C_{d-2}(\Delta; \mathbb{Z}) \rightarrow \cdots$$

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

Define

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

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

Theorem (DKM, pp '11)

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

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

Theorem (DKM, pp ’11)

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

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

Corollary

\[ |K(\Delta)| \text{ is the torsion-weighted number of } d\text{-dimensional spanning trees of } \Delta. \]

Proof.

\[ |K(\Delta)| = |(\mathbb{Z}^t)/L_{\Gamma}| = |\det L_{\Gamma}|, \text{ which counts (torsion-weighted) spanning trees.} \]
What does it look like?

\[ K(\Delta) := \ker \partial_{d-1}/\text{im } L_{d-1} \subseteq \mathbb{Z}^m \]
What does it look like?

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

- Put integers on \((d - 1)\)-faces of \(\Delta\). Orient faces arbitrarily.
  - \(d = 2\): flow; \(d = 3\): circulation; etc.
What does it look like?

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

- Put integers on \((d-1)\)-faces of \(\Delta\). Orient faces arbitrarily.
  - \(d = 2\): flow; \(d = 3\): circulation; etc.
- Conservative flow
What does it look like?

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

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

\[ \begin{array}{c}
5 \\
6 \\
\downarrow \\
4 \\
\downarrow \\
1 \\
2 \\
\end{array} \]
What does it look like?

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

- Put integers on \((d - 1)\)-faces of \(\Delta\). 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?

\[ K(\Delta) := \ker \partial_{d-1} / \text{im } L_{d-1} \subset \mathbb{Z}^m \]

- Put integers on \((d - 1)\)-faces of \(\Delta\). 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

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

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

$K(\Delta) := \ker \partial \frac{d}{d-1} \sim \frac{im L}{d-1}$

Proof.

$\triangleright K(\Delta)$ is generated by boundaries of facets $\partial F$.

$\triangleright$ In a sphere, the Laplacian of a ridge shows if facets $F, G$ are adjacent, then $\partial F \equiv \pm \partial G \pmod{im L}$.

$\triangleright$ So $K(\Delta)$ has a single generator, so it is cyclic.

$\triangleright |K(\Delta)|$ is the number of spanning trees, and there is one tree for every facet (remove that facet for the tree).
Example: Spheres

Theorem

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

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

Proof.

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

\[ K(\Delta) \equiv \mathbb{Z}_n \]
Example: Spheres

Theorem

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

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

Proof.

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

**Theorem**

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

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

**Proof.**

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

**Theorem**

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

\[
K(\Delta) := \ker \partial_{d-1} / \text{im } L_{d-1}
\]

**Proof.**

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