University of Wisconsin
SCHOOL OF MEDICINE
AND PUBLIC HEALTH

Hodge Theory for Medical Image

Moo K. Chung
University of Wisconsin-Madison
www.stat.wisc.edu/~mchung

References

* Lee et al. 2014 MICCAI 8675:297-304
http://pages.stat.wisc.edu/~mchung/papers/lee.2014.MICCAI.pdf
Introduction of Hodge Laplacian for graphs/networks

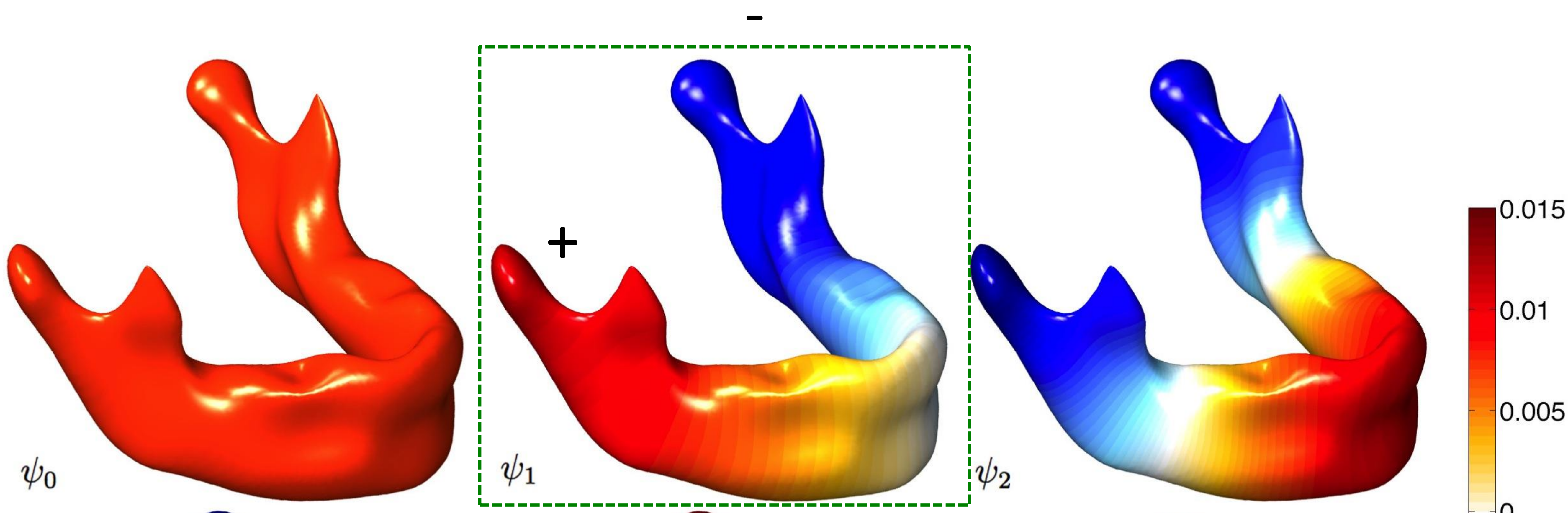
Lee et al. 2018 ISBI 20-23
http://pages.stat.wisc.edu/~mchung/papers/lee.2018.ISBI.pdf
Introduction of cycles

Lee et al. 2019 LNCS 11382:110-122
http://pages.stat.wisc.edu/~mchung/papers/lee.2018.CTIC.pdf
Distance between cycles

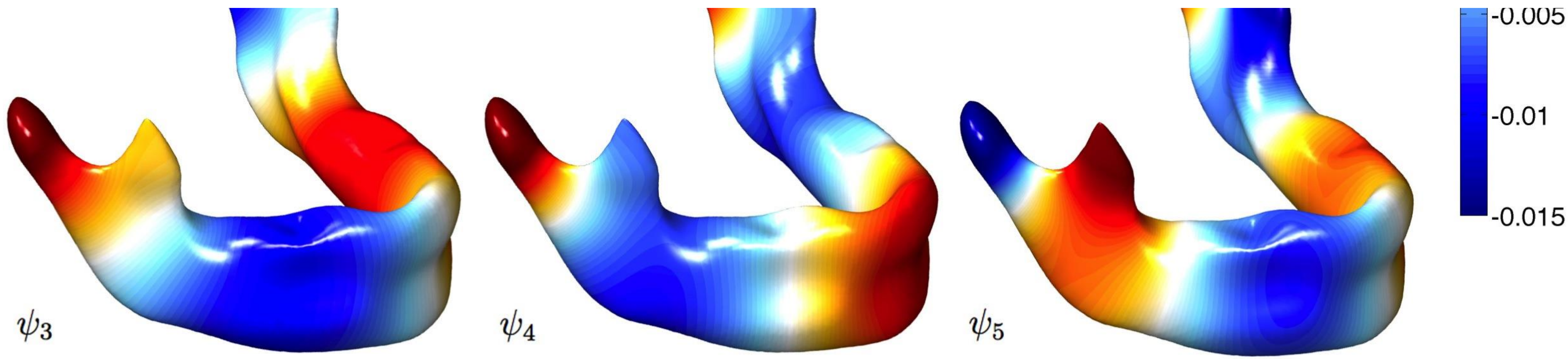
* Lee et al. 2019 MICCAI 11767:674-682
http://pages.stat.wisc.edu/~mchung/papers/lee.2019.MICCAI.pdf
Co-identification of cycles through the Stiefel optimization

* Songdechakraiwut et al. 2020 arXiv:2012.00675
https://arxiv.org/pdf/2012.00675.pdf
Wasserstein distance between cycles

Motivation: Laplace-Beltrami eigenfunctions on a mandible



Why the maximum and minimum occurs at the extreme?

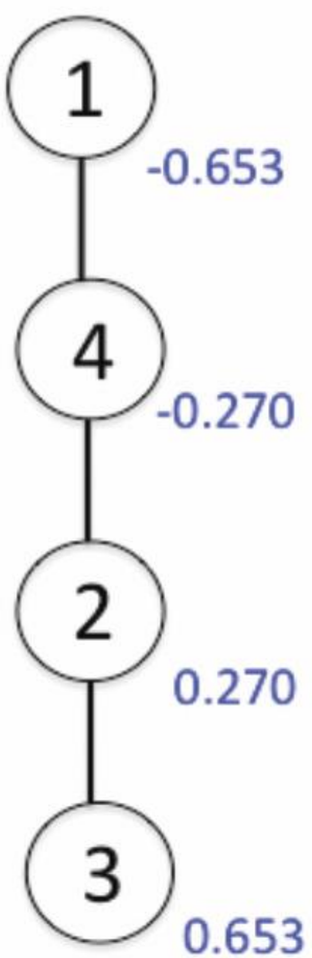


$$\Delta\psi_j = \lambda_j \psi_j$$

Seo et al. 2010 MICCAI 6363:505-512

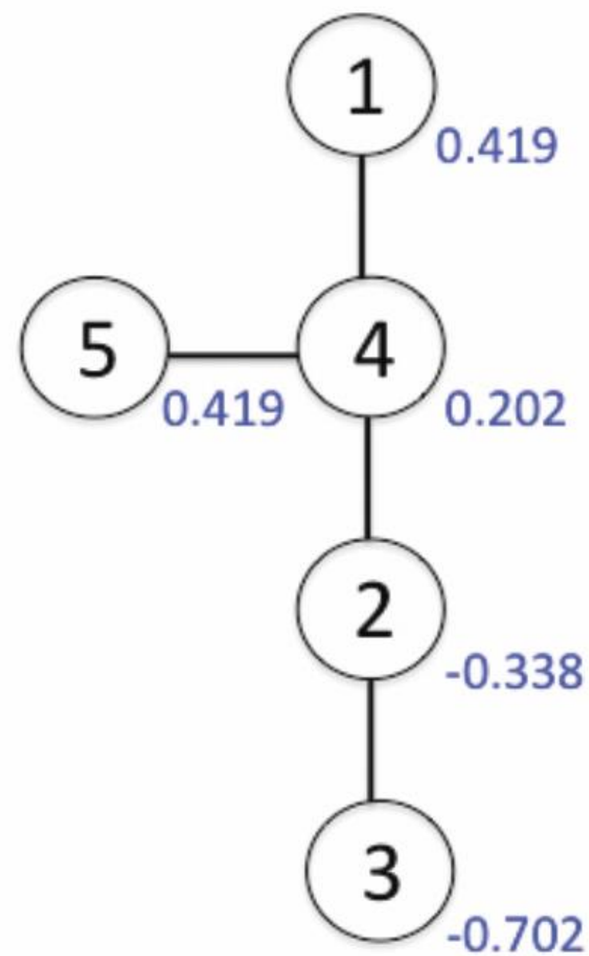
Graph Laplacian

$$L = (l_{ij}) \qquad l_{ij} = \begin{cases} -w_{ij}, & i \sim j \\ \sum_{i \neq j} w_{ij}, & i = j \\ 0, & \text{otherwise} \end{cases}$$



$$W = \begin{matrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{matrix}$$

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



$$W = \begin{matrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{matrix}$$

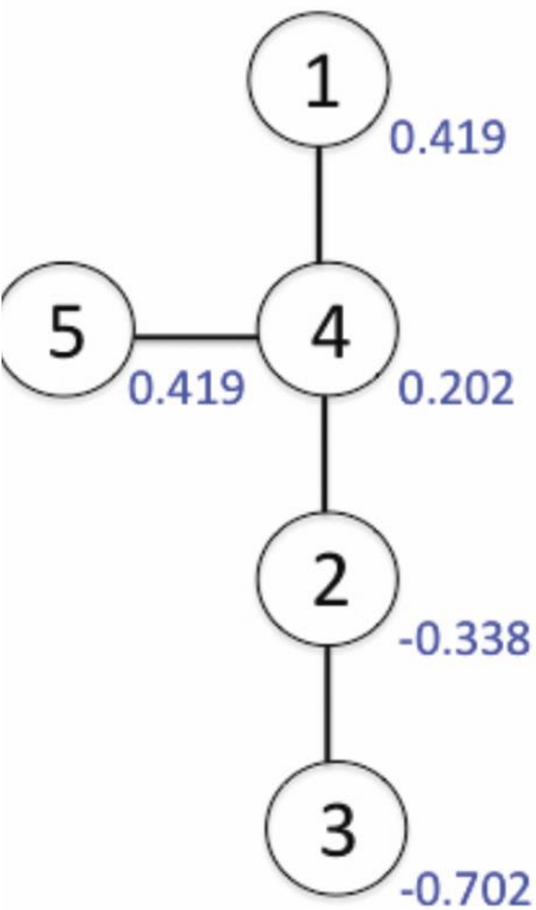
$$L = \begin{matrix} 1 & 0 & 0 & -1 & 0 \\ 0 & 2 & -1 & -1 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 3 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{matrix}$$

Graph Laplacian from adjacency matrix

$$L = D - A$$

A: adjacency matrix

$$D = diag(deg(1), \cdots, deg(p))$$



$$W = \begin{matrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{matrix}$$

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

Graph Laplacian from incidence matrix

(*i,j*)-th entry = 1 if $\tau_i \subset \sigma_j$

Sign depends on the orientation of τ_i

$$L = MM^\top$$

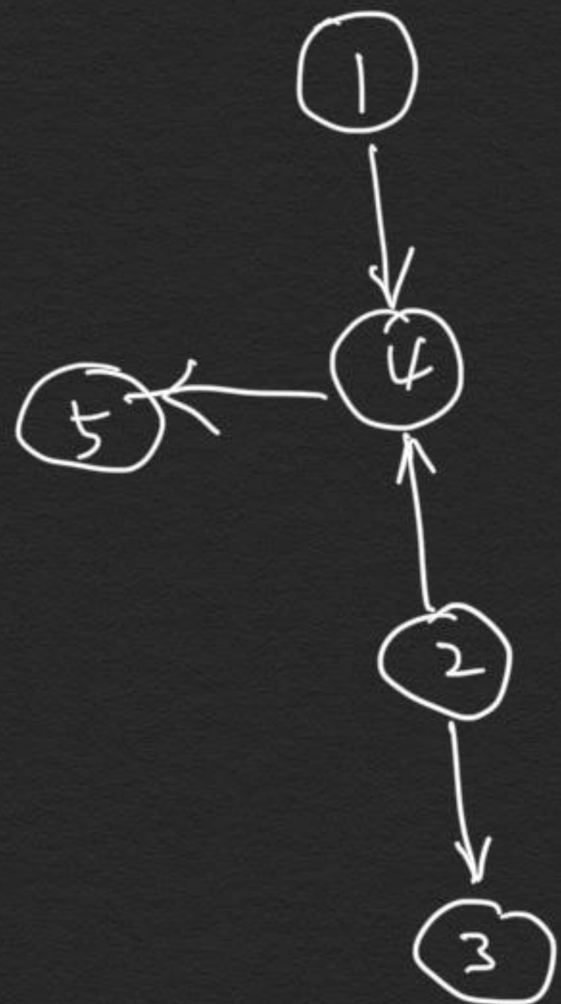
M # of *edges* σ_j

# of vertices τ_i

$$\begin{array}{|ccccccc|}
\hline
1 & 0 & 1 & \dots & 1 & 0 & 1 \\
\cdot & & & & & \cdot & \\
\cdot & & & & & \cdot & \\
\cdot & & & \boxed{(i,j)} & & & \cdot \\
1 & 1 & & & & 0 & 0 \\
0 & 1 & 0 & \dots & 0 & 1 & 1 \\
\hline
\end{array}$$

For most graphs,
# of vertices < # of edges

Graph Laplacian from incidence matrix



M	12	13	14	15	23	24	25	34	35	45
1	0	0	-1	0	0	0	0	0	0	0
2	0	0	0	0	-1	-1	0	0	0	0
3	0	0	0	0	1	0	0	0	0	0
4	0	0	1	0	0	1	0	0	0	-1
5	0	0	0	0	0	0	0	0	0	1

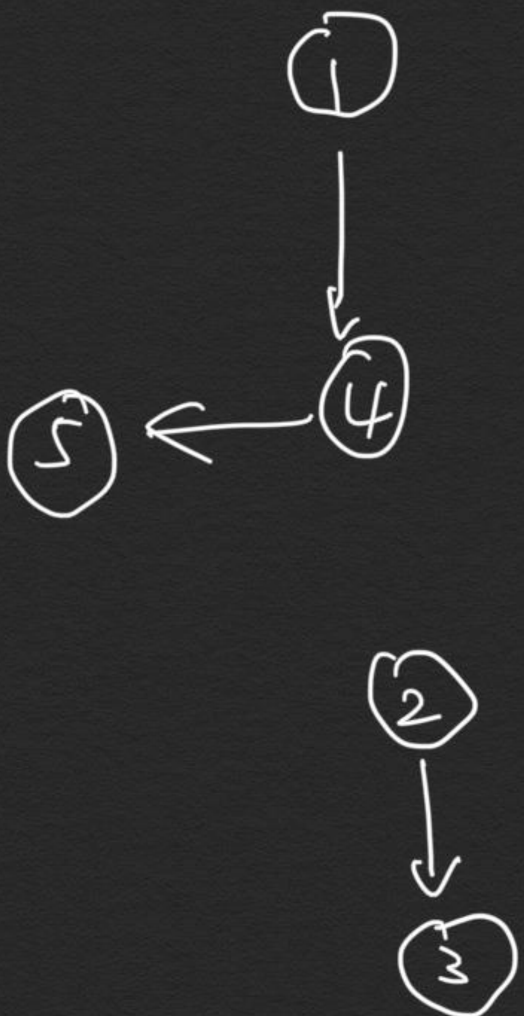
Lexical indexing

$$MM^T = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 \\ 0 & 2 & -1 & -1 & 0 \\ 0 & -1 & 1 & \boxed{0} & 0 \\ -1 & 1 & 0 & 3 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

Multiply
3rd and 4th rows

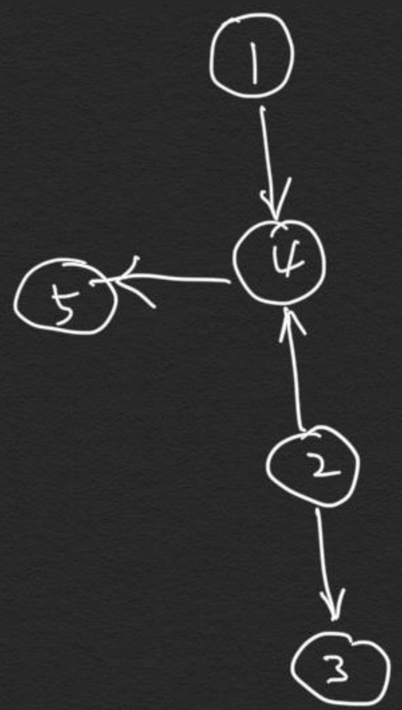
Laplacian invariant under indexing (parameterization)

How the incidence matrix changes over edge deletion

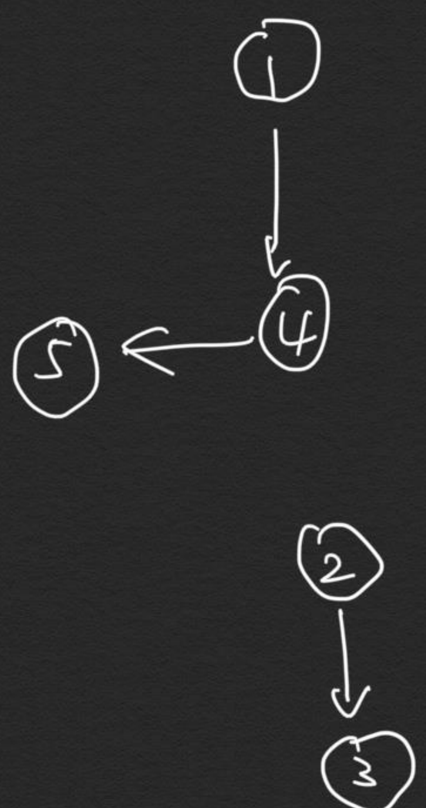


M	12	13	14	15	23	24	25	34	35	45
1	0	0	-1	0	0	0	0	0	0	0
2	0	0	0	0	-1	0	0	0	0	0
3	0	0	0	0	1	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0	-1
5	0	0	0	0	0	0	0	0	0	1

How the Laplacian changes over edge deletion



$$MM^T = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 \\ 0 & 2 & -1 & -1 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 3 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$



$$MM^T = \begin{array}{c} \\ 1 \\ 4 \\ 5 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 4 & 5 & 2 & 3 \end{array} \\ \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \end{array}$$

Eigenvectors of Laplacian

$$L\mathbf{f}_i = \lambda_i \mathbf{f}_i$$

$$0 = \lambda_0 \le \lambda_1 \le \cdots \le \lambda_p$$

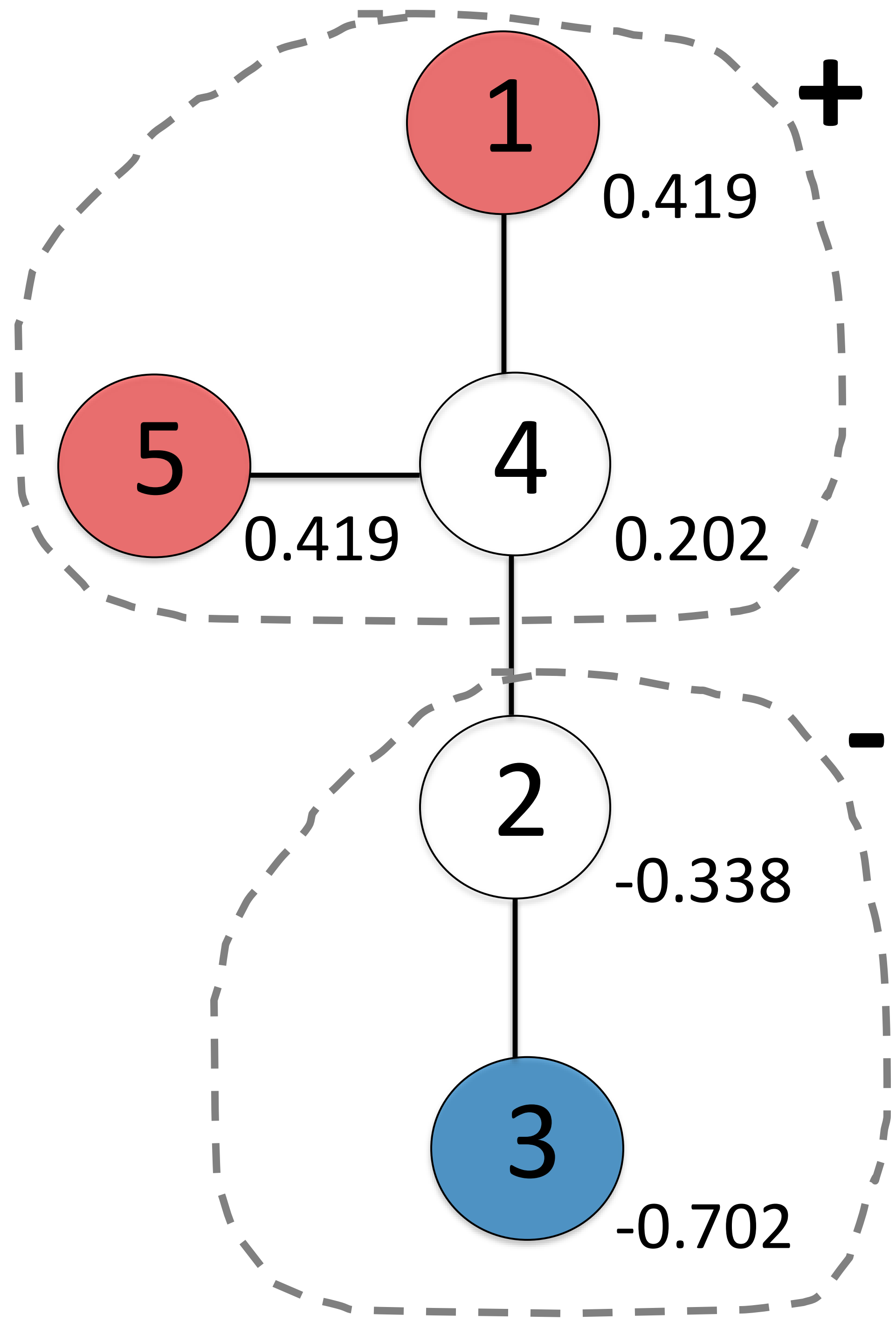
What is $\mathbf{f}_0$?

Fiedler vector $L\mathbf{f}_1 = \lambda_1 \mathbf{f}_1$

Hilbert nodal line theorem: The Fiedler vector partitions the node set into exactly two sign domains.

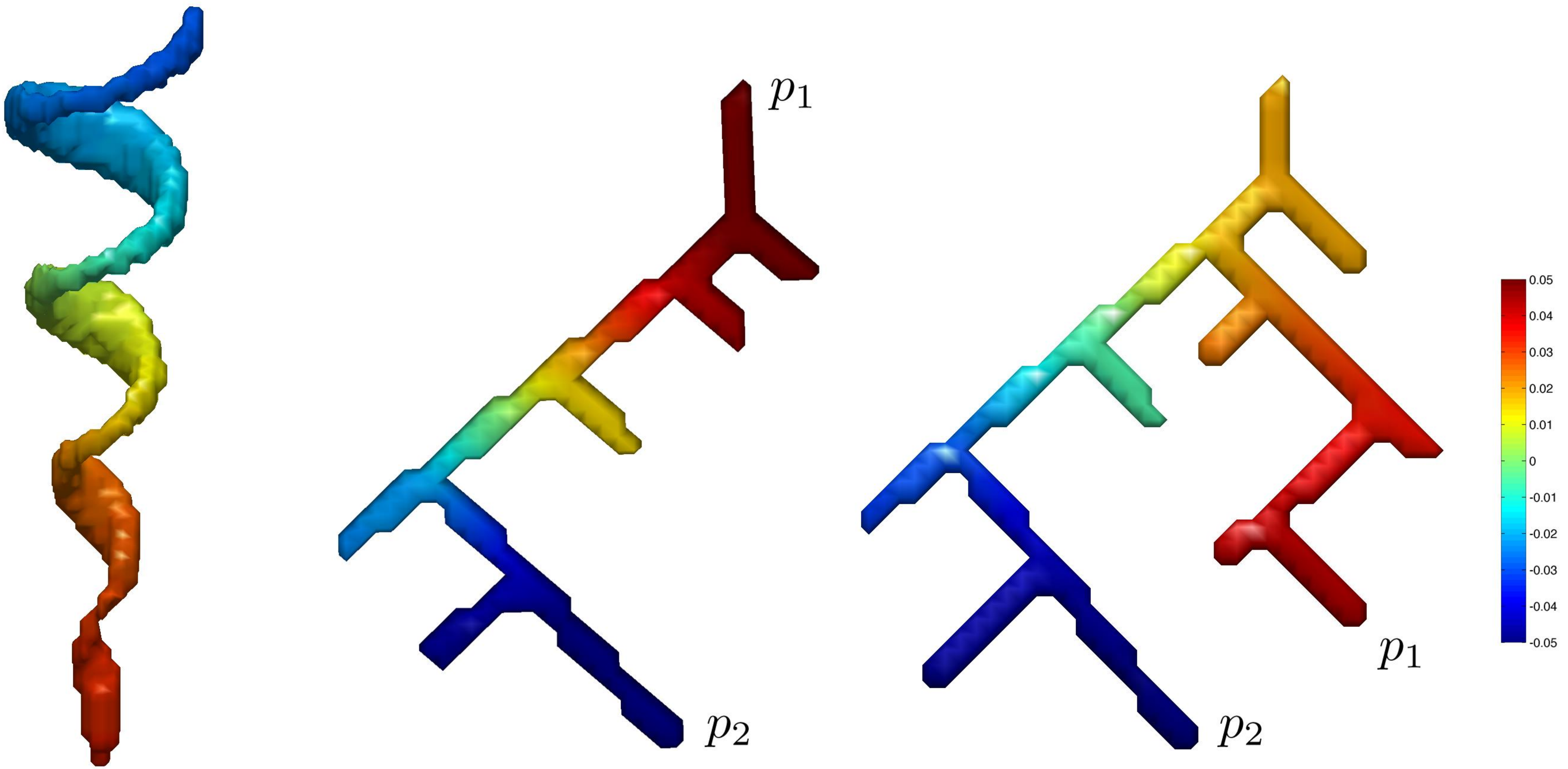
Hot spot at the positive domain.
Cold spot at the negative domain.

Tlusty 2007: The hot/cold spots have to occur at the boundary.

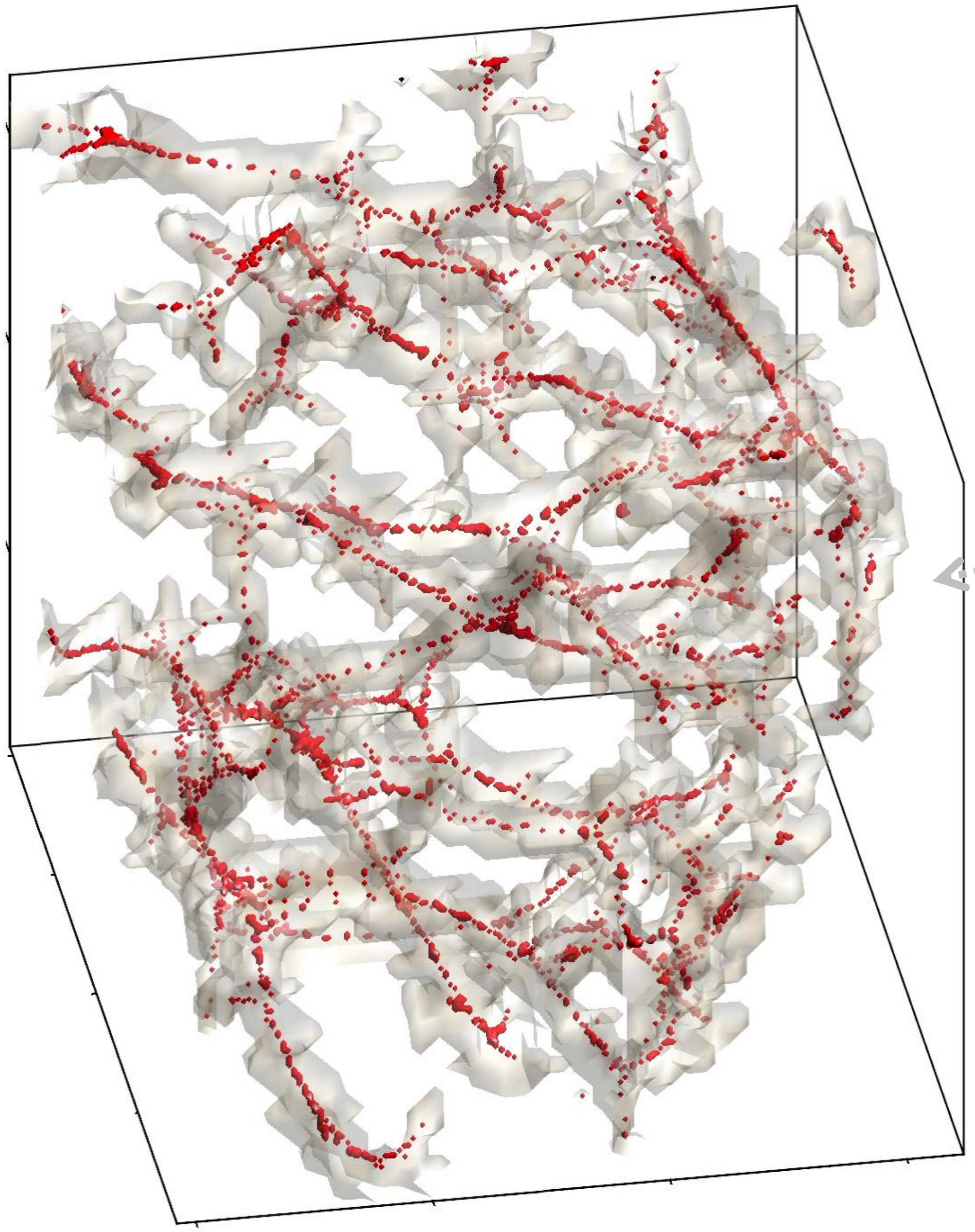


<u>*Chung et al. 2011 MLMI 7009:225-232*</u>

Fiedler vector on trees $L\mathbf{f}_1 = \lambda_1 \mathbf{f}_1$

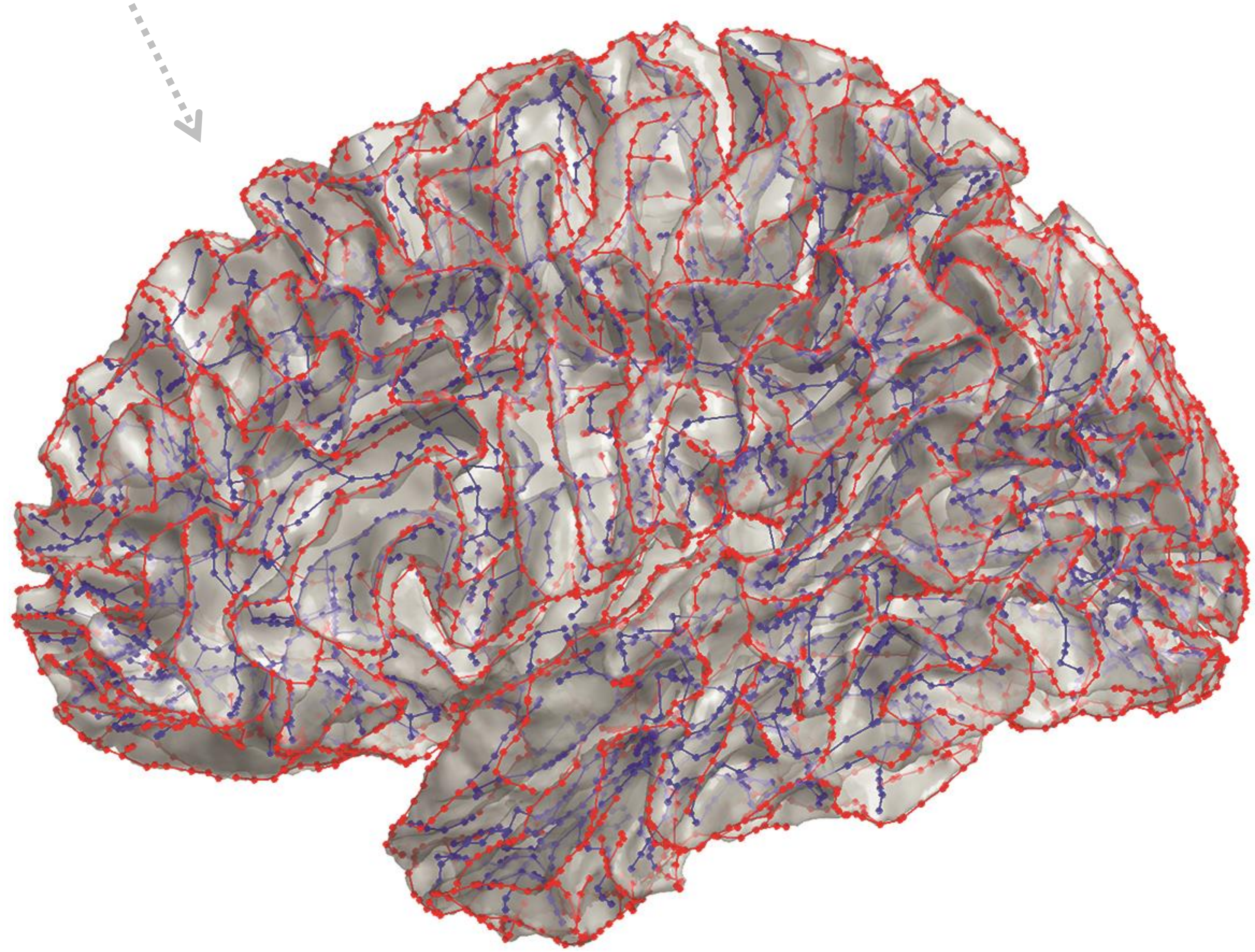


Fiedler's vector will elongate data → clustering



Blood vessels in lung CT image
Chung et al. 2018 EMBC 5101-5104

Research problem: Determining the length of extremely complicated tubular structures and networks (lung blood vessel trees, structural brain networks, sulcal trees) in medical imaging.



Chung & Ombao 2021 Biometrika Discussion on TDA

Multiplicity of eigenvalues

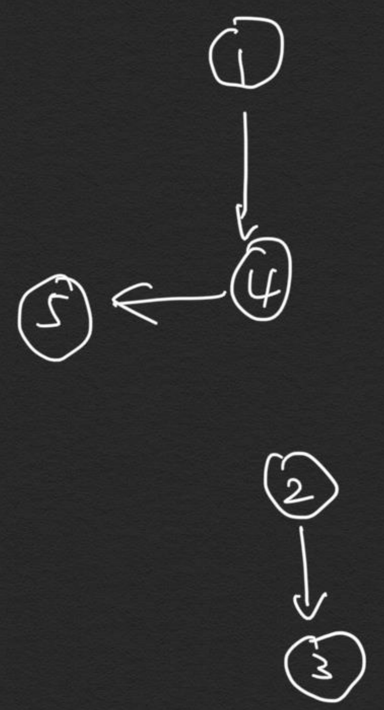
Multiplicity m of eigenvalue λ :

$$L\mathbf{v}_1 = \lambda\mathbf{v}_1, \cdots, L\mathbf{v}_m = \lambda\mathbf{v}_m$$

Theorem: The multiplicity of eigenvalue $\lambda_0 = 0$ is the number of connected components: 0-th Betti number β_0.

Equivalently $\beta_0 = p - rank(L)$

Multiplicity of eigenvalues



$$
\begin{array}{c|c}
 & \begin{array}{ccccc} 1 & 4 & 5 & 2 & 3 \end{array} \\
\begin{array}{c} 1 \\ 4 \\ 5 \\ 2 \\ 3 \end{array} &
\left[\begin{array}{ccccc} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{array}\right]
\end{array}
$$

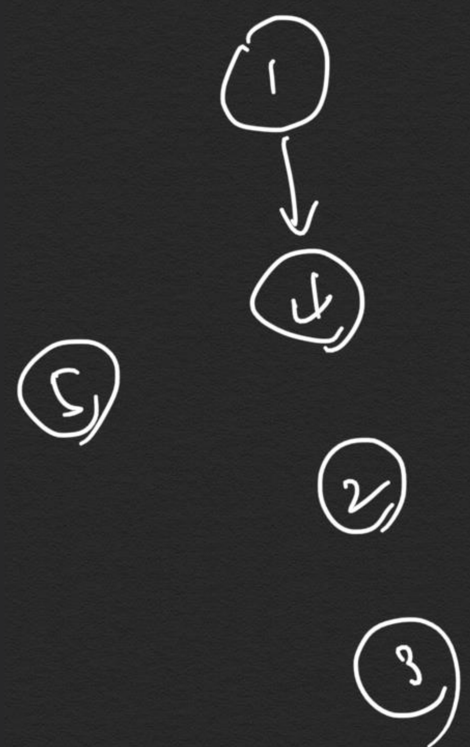
```
>> [F, D] = eig(L)

F =
         0   -0.5774   -0.7071         0    0.4082
         0   -0.5774    0.0000         0   -0.8165
         0   -0.5774    0.7071         0    0.4082
   -0.7071         0         0   -0.7071         0
   -0.7071         0         0    0.7071         0

D =
         0         0         0         0         0
         0    0.0000         0         0         0
         0         0    1.0000         0         0
         0         0         0    2.0000         0
         0         0         0         0    3.0000
```

$rank(L) = 3 \qquad \beta_0 = 2$

Theorem: $\beta_0 = p - rank(L)$

Proof. $$rank(L) = rank(MM^\top) = rank(M^\top)$$

$$kerL = \{\sigma : L\sigma = 0\} = \{\sigma : M^\top\sigma = 0\}$$



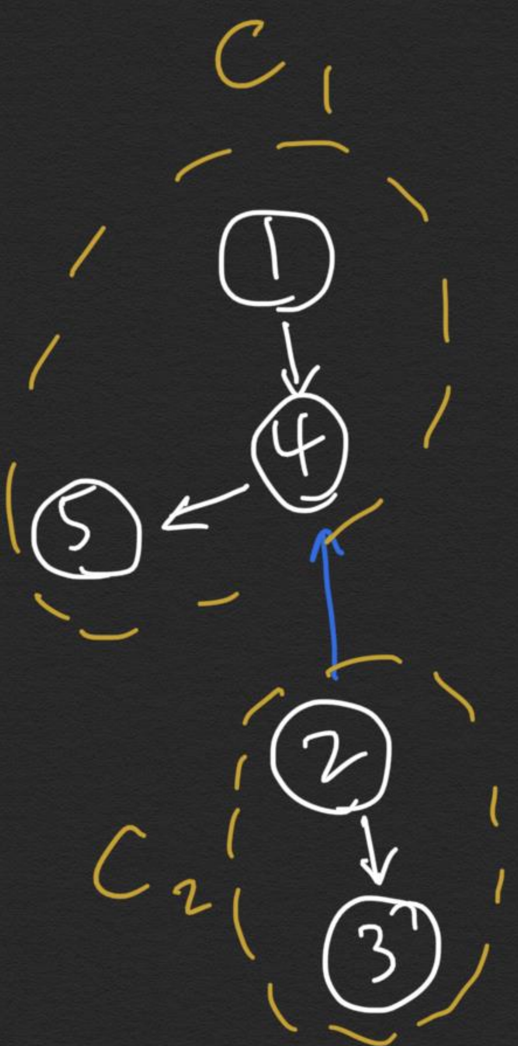
M	...	14	...
1		-1	
5			
4		1	
2			
3			

$$\sigma = [a, a, b, c, d]^\top$$

$$rank(kerL) = 4$$
$$rank(L) = 1$$

Graph with 2 disconnected components

$$kerL = \{\sigma : L\sigma = 0\} = \{\sigma : M^\top \sigma = 0\}$$



M	14	15	45	24	23	12	13	25	34	35
1										
5		C_1		0	0			0		
4				0						
2				0						
3		0		0	C_2			0		

$$rank(kerL) = \beta_0 = 2$$

$$rank(L) = p - 2$$

$$\sigma = [a, a, a, b, b]^\top$$

Rank-nullity theorem for graph Laplacian

$$kerL = \{\sigma : L\sigma = 0\} = \{\sigma : M^\top \sigma = 0\}$$

$$imgL = \{L\sigma : \sigma \in \mathbb{R}^p\}$$

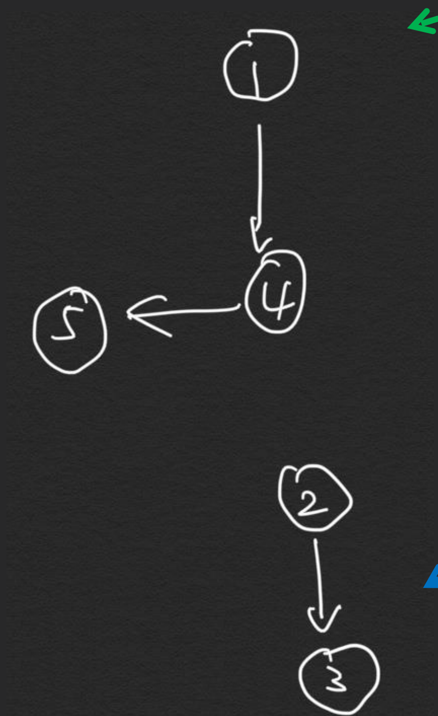
$$rank(imgL) + rank(kerL) = p$$

$rank(L)$ β_0

How to find the connected components?

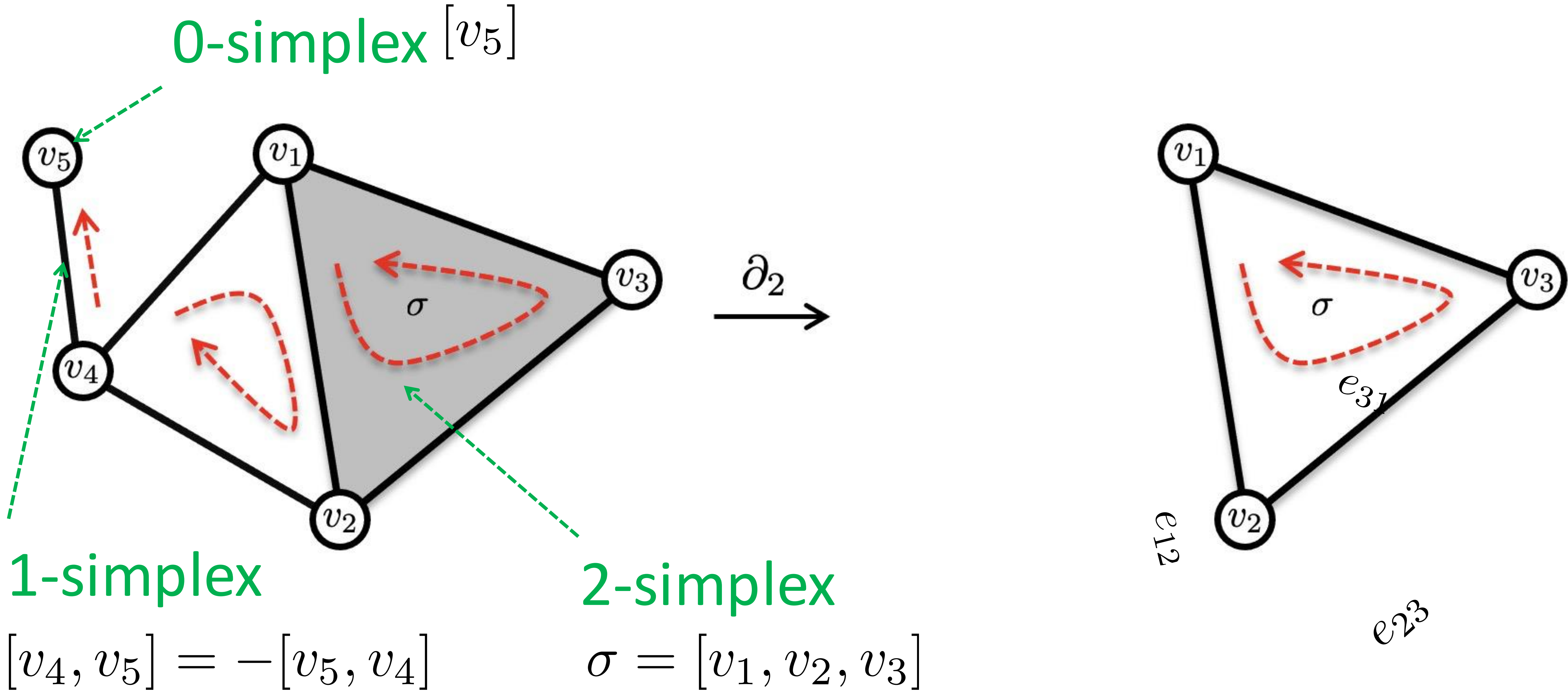
$$\ker L = \{\sigma : L\sigma = 0\} = \{\sigma : M^\top \sigma = 0\}$$

The kernel space is spanned by the eigenvectors corresponding to zero eigenvalue




```
>>  [F, D] = eig(L)

F =
        0   -0.5774   -0.7071         0    0.4082
        0   -0.5774    0.0000         0   -0.8165
        0   -0.5774    0.7071         0    0.4082
  -0.7071         0         0   -0.7071         0
  -0.7071         0         0    0.7071         0

D =
        0         0         0         0         0
        0    0.0000         0         0         0
        0         0    1.0000         0         0
        0         0         0    2.0000         0
        0         0         0         0    3.0000
```

k-simplex

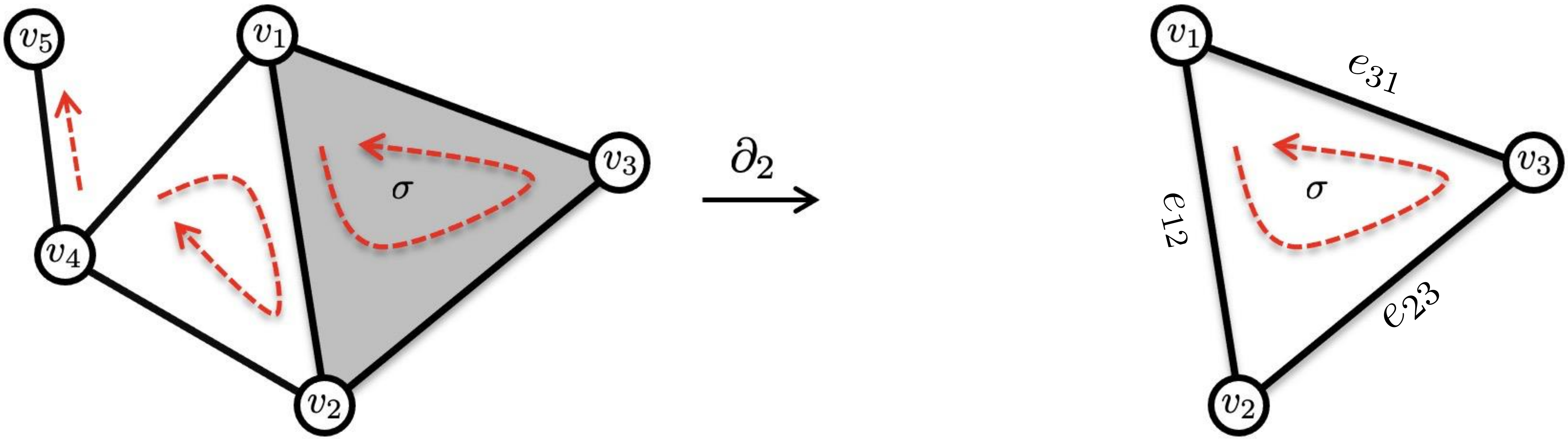


C_k :collection of *k*-simplexes (*or simplices*)

Boundary operators ∂_k

∂_k Removes the filled-in interior of *k*-simplexes

$\partial_k : C_k \to C_{k-1}$



$$\partial_2 \sigma = [v_1, v_2] + [v_2, v_3] + [v_3, v_1]$$

$$\partial_1 \partial_2 \sigma = v_2 - v_1 + v_3 - v_2 + v_1 - v_3 = 0$$

$$\partial_1 \partial_2 \sigma = v_2 - v_1 + v_3 - v_2 + v_1 - v_3 = 0$$

Theorem. $\partial_{k-1} \partial_k \sigma = 0$

Example. Boundary of boundary of a filled-in tetrahedron = edges

→ Does the sum of edges vanish?

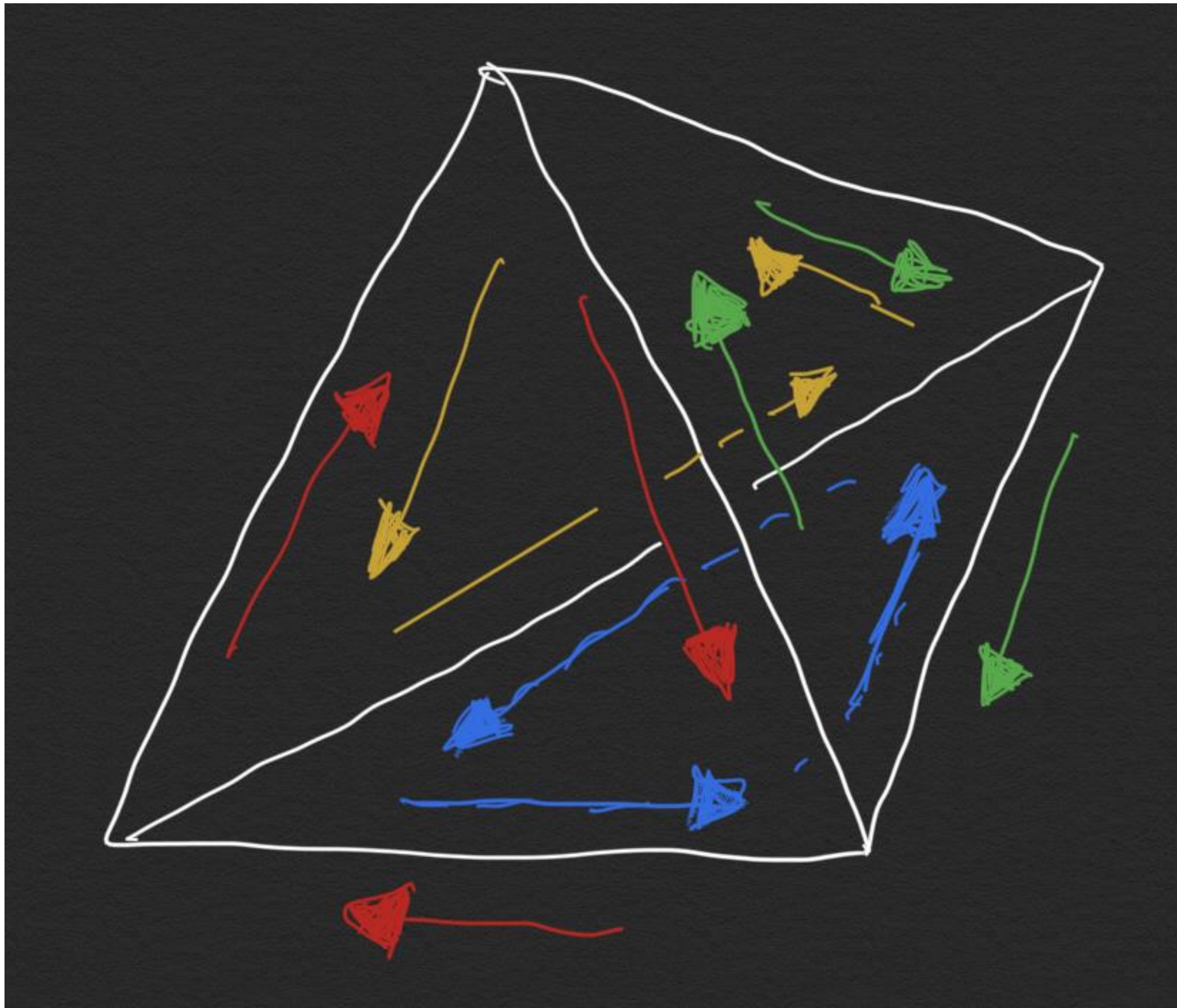
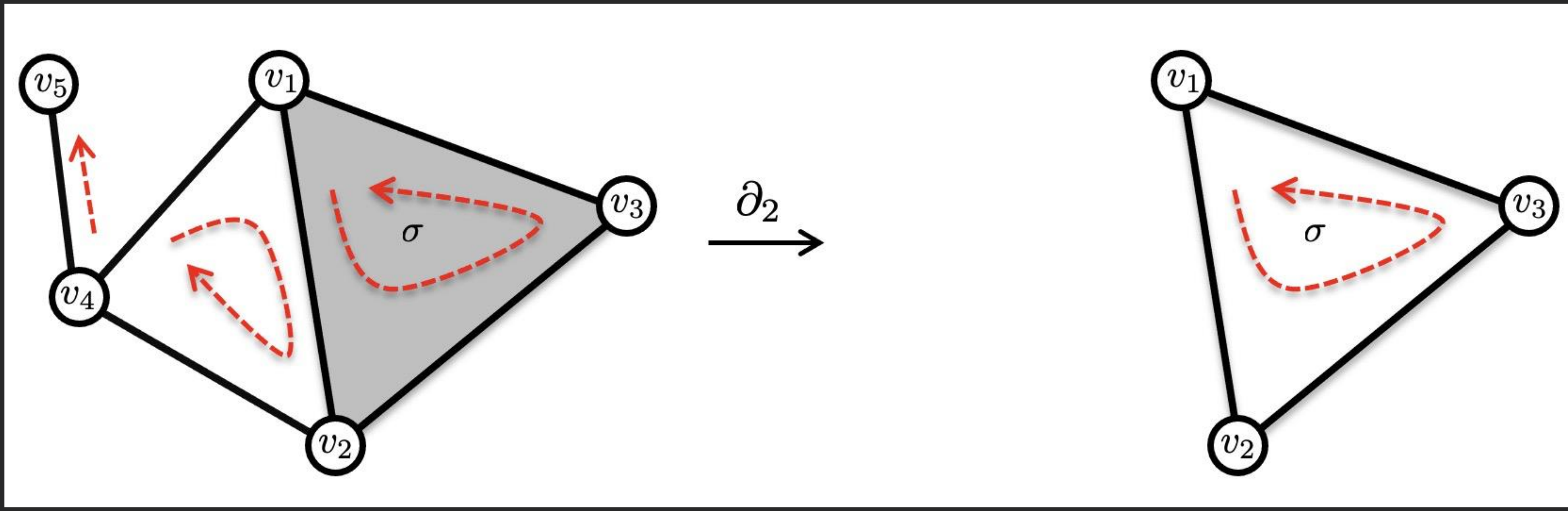


Image of boundary operator

$$img\ \partial_{k+1} = \{\partial_{k+1}\sigma : \sigma \in C_{k+1}\}$$

collection of *k*-boundaries

$\partial_2\sigma = [v_1, v_2] + [v_2, v_3] + [v_3, v_1]$ is 1-boundary



Kernel of boundary operator

$$ker\ \partial_k = \{\sigma \in C_k : \partial_k \sigma = 0\}$$

collection of *k*-cycles

$$\partial_1 \partial_2 \sigma = v_2 - v_1 + v_3 - v_2 + v_1 - v_3 = 0$$

$$\partial_2 \sigma = [v_1, v_2] + [v_2, v_3] + [v_3, v_1]$$ is 1-cycle

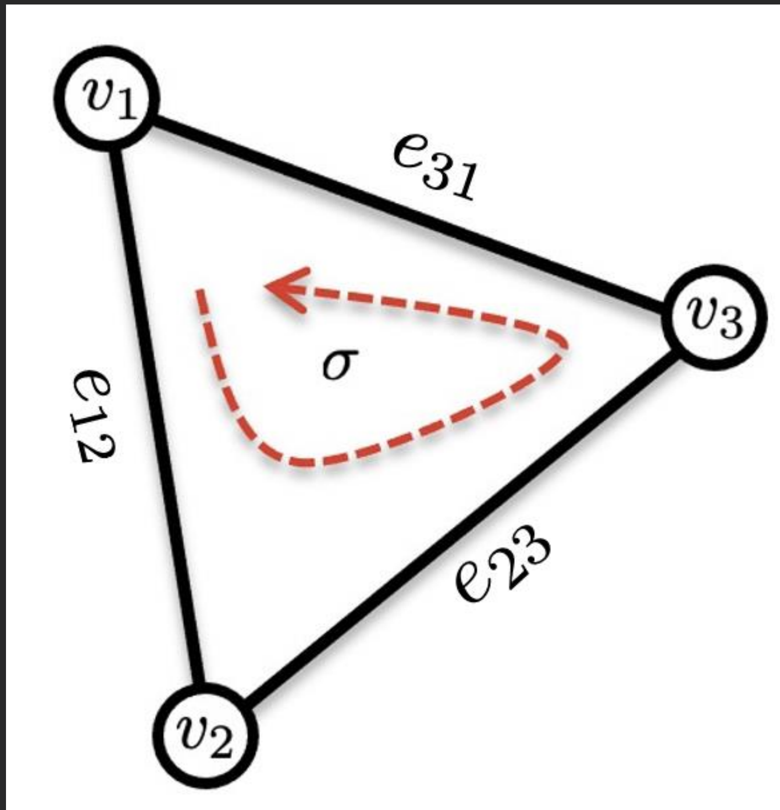


Image of boundary operator

From theorem $\partial_k \partial_{k+1} \sigma = 0$
boundary $\partial_{k+1}\sigma$ is always a cycle.

$$ker \partial_k \supset img \partial_{k+1}$$

Set of cycles   Set of boundaries

Quotient space:

$$H_k = ker \partial_k / img \partial_{k+1}$$

Total number of algebraically independent cycles (# of basis).

$$\beta_k = rank(ker \partial_k) - rank(img \partial_{k+1})$$

Lee et al. 2014 (MICCAI 8675:297-304) *explains* matrix $\boldsymbol{\partial}_k$

of *k*-dimensional simplices σ_j

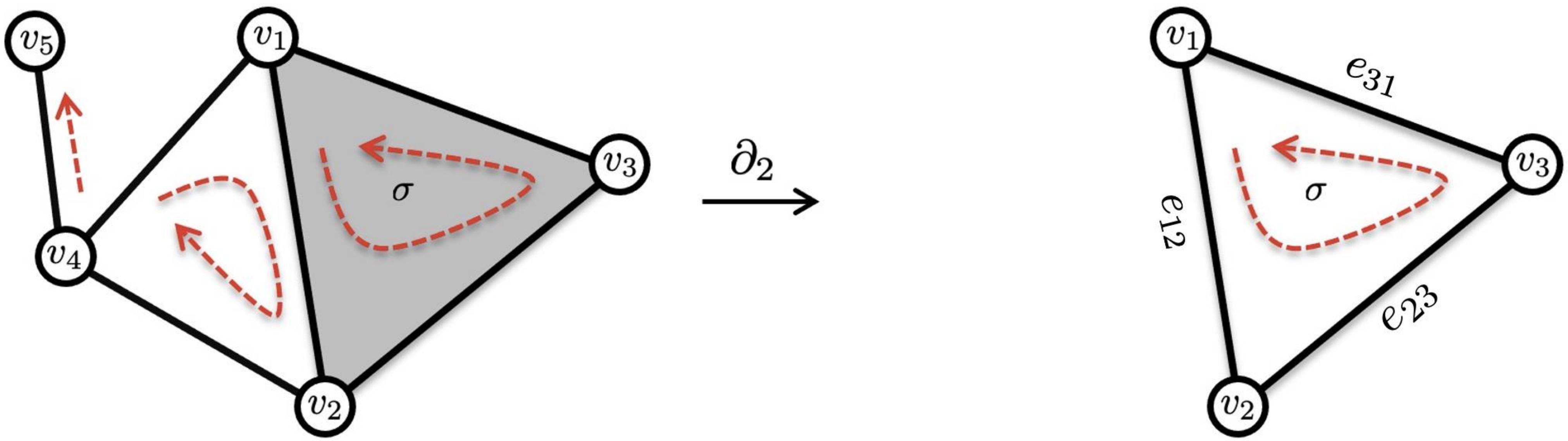
of (*k-1*)-dimensional simplices τ_i

$$
\begin{array}{cccccccc}
1 & 0 & 1 & \ldots & 1 & 0 & & 1 \\
. & & & & & . & & \\
. & & & & & . & & \\
. & & & & & . & & \\
1 & 1 & & & & 0 & 0 & \\
0 & 1 & 0 & \ldots & & 0 & 1 & -1
\end{array}
$$

(*i*,*j*)-th entry = 1 if $\tau_i \subset \sigma_j$

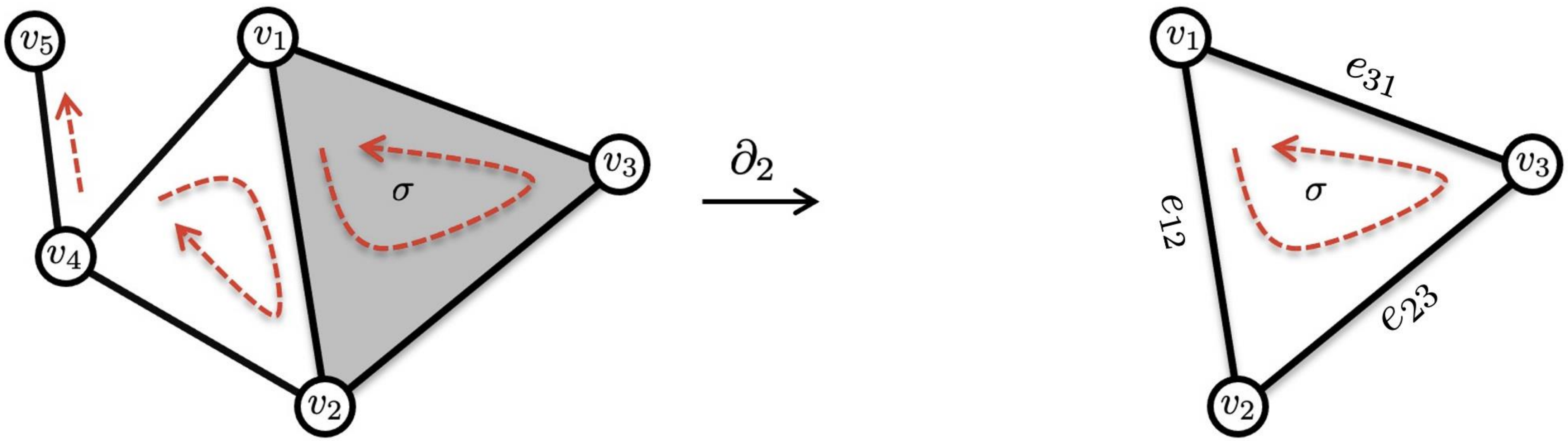
Sign depends on the orientation of τ_i

Boundary matrix $\boldsymbol{\partial}_0$



$$\boldsymbol{\partial}_0 = \begin{array}{cc} & \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ 0 & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Boundary matrix $\boldsymbol{\partial}_1$



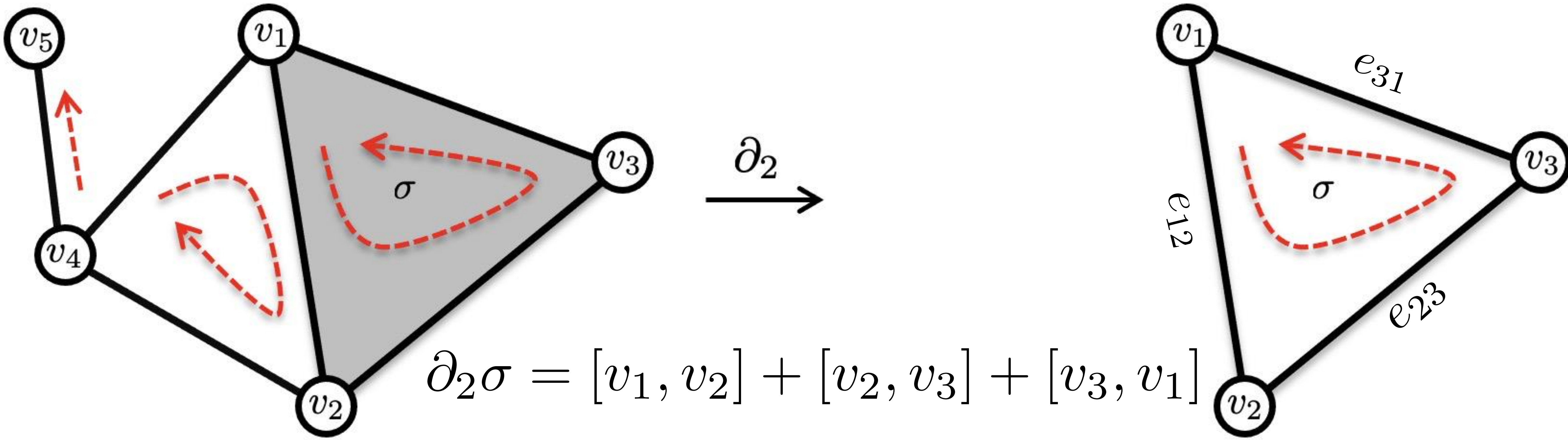
$$\partial_1 \partial_2 \sigma = v_2 - v_1 + v_3 - v_2 + v_1 - v_3 = 0$$

$$\boldsymbol{\partial}_1 = \begin{array}{cc} & \begin{array}{cccccc} & & \sigma & & & \\ e_{12} & e_{23} & e_{31} & e_{24} & e_{41} & e_{45} \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} & \begin{pmatrix} -1 & 0 & 1 & 0 & 1 & 0 \\ 1 & -1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Boundary operators ∂_k

∂_k Removes the filled-in interior of *k*-simplexes

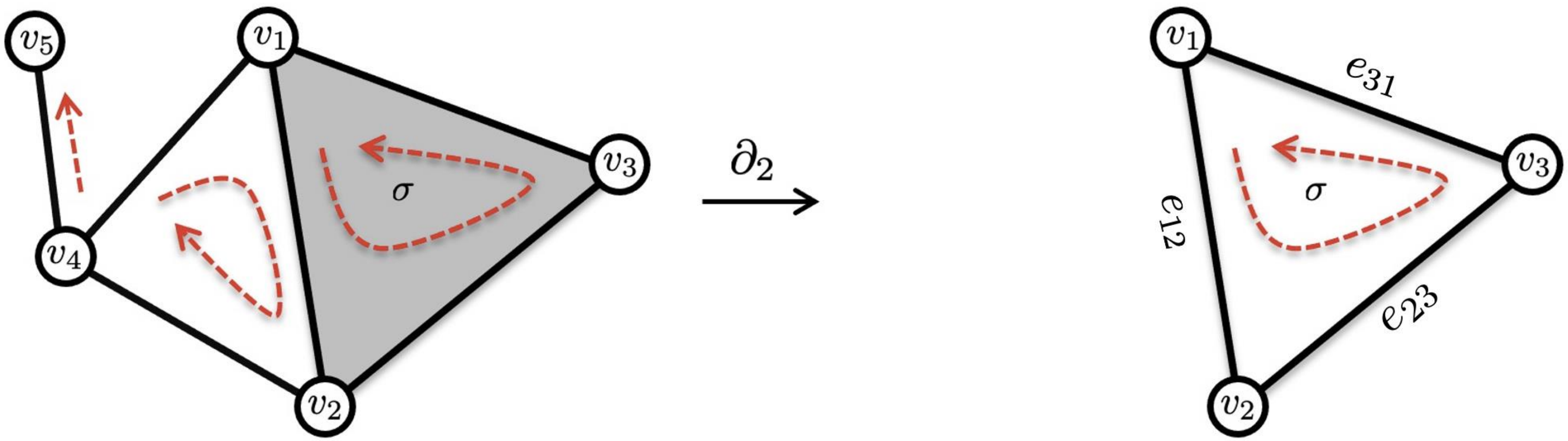
$$\partial_k : C_k \to C_{k-1}$$



$$\partial_2\sigma = [v_1, v_2] + [v_2, v_3] + [v_3, v_1]$$

$$\boldsymbol{\partial}_2 = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccc} e_{12} & e_{23} & e_{31} & e_{24} & e_{41} & e_{45} \end{array} \\ \begin{pmatrix} -1 & 0 & 1 & 0 & 1 & 0 \\ 1 & -1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

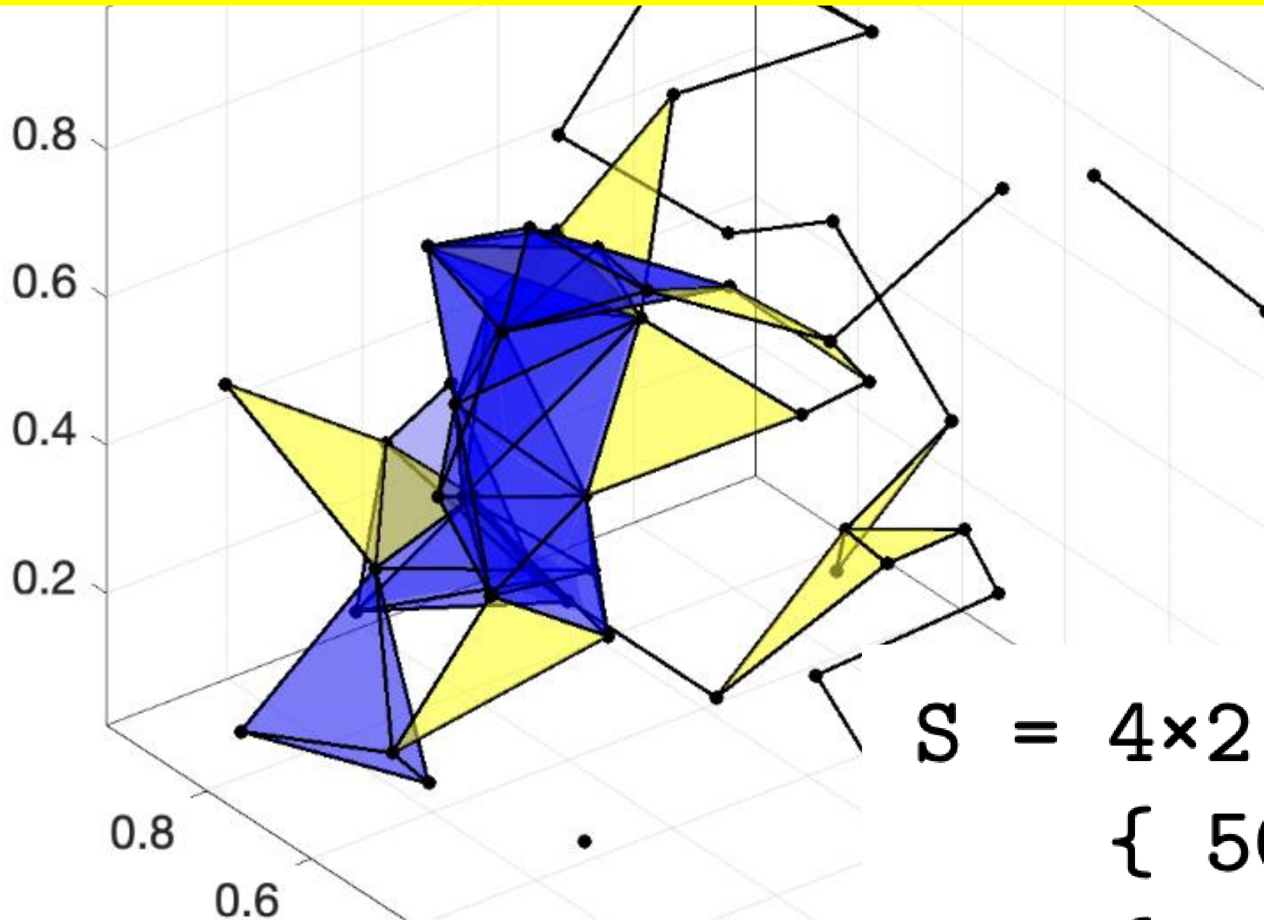
Boundary matrix $\boldsymbol{\partial}_2$



$$\partial_2 \sigma = [v_1, v_2] + [v_2, v_3] + [v_3, v_1]$$

$$\boldsymbol{\partial}_2 = \begin{array}{c} \\ e_{12} \\ e_{23} \\ e_{31} \\ e_{24} \\ e_{41} \\ e_{45} \end{array} \begin{array}{c} \sigma \\ \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \end{array}$$

Numerical implementation: PH-STAT



Connectivity

Functional data over simplex

```
S = 4×2 cell array
    { 50×1 double}     { 50×1 double}
    {101×2 double}     {101×1 double}
    { 73×3 double}     { 73×1 double}
    { 27×4 double}     { 27×1 double}
```

ex. 3-node connectivity

```
[S{3,1} S{3,2}]
ans =   1     5    13     0.2932
        1     5    16     0.2579
        1     5    26     0.2766
                 . . .
```

Rank-nullity theorem for boundary matrix

$\boldsymbol{\partial}_k$

Smith normal form $S(\boldsymbol{\partial}_k)$

$$\begin{array}{c|c}
 & \text{\# of } k\text{-D simplices } \sigma_j \\
\hline
\text{\# of } (k\text{-}1)\text{-D simplices } \tau_i &
\begin{bmatrix}
1\ 0\ 1 \ldots & 1\ 0\ 1 \\
. & . \\
. & . \\
. & . \\
1\ 1 & 0\ 0 \\
0\ 1\ 0 \ \ldots & 0\ 1\ 1
\end{bmatrix}
\end{array}
\quad \xrightarrow{\text{Gaussian elimination}} \quad
\begin{array}{cc|c}
\overleftrightarrow{rank(\boldsymbol{\partial}_k)} & \overleftrightarrow{rank(ker\boldsymbol{\partial}_k)} & \\
\begin{bmatrix}
1\ 0\ \ldots & \\
. & \\
. \quad 1\ 0 & \\
0\ 0\ \ldots\ 0\ 1 & \\
\hline
0\ 0 & \\
0\ 0\ 0\ \ldots &
\end{bmatrix}
&
\begin{bmatrix}
0\ 0 \\
. \\
. \\
\\
\hline
0\ 0 \\
0\ 0
\end{bmatrix}
&
\begin{array}{l}
rank(\boldsymbol{\partial}_k^\top) \\
\\
rank(ker\boldsymbol{\partial}_k^\top)
\end{array}
\end{array}$$

(i,j)-th entry = 1 if $\tau_i \subset \sigma_j$
Sign depends on the orientation of τ_i

$$\beta_k = rank(ker\boldsymbol{\partial}_k) - rank(\boldsymbol{\partial}_{k+1})$$

Smith normal form

$$\boldsymbol{\partial}_1 = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccc} e_{12} & e_{23} & e_{31} & e_{24} & e_{41} & e_{45} \end{array} \\ \begin{pmatrix} -1 & 0 & 1 & 0 & 1 & 0 \\ 1 & -1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Gaussian elimination

$$\mathcal{S}(\boldsymbol{\partial}_1) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$rank(\boldsymbol{\partial}_1)$

$rank(ker\boldsymbol{\partial}_1)$

$rank(\boldsymbol{\partial}_1^\top) = 4$

$rank(ker\boldsymbol{\partial}_1^\top) = 1$

# of nodes = 5

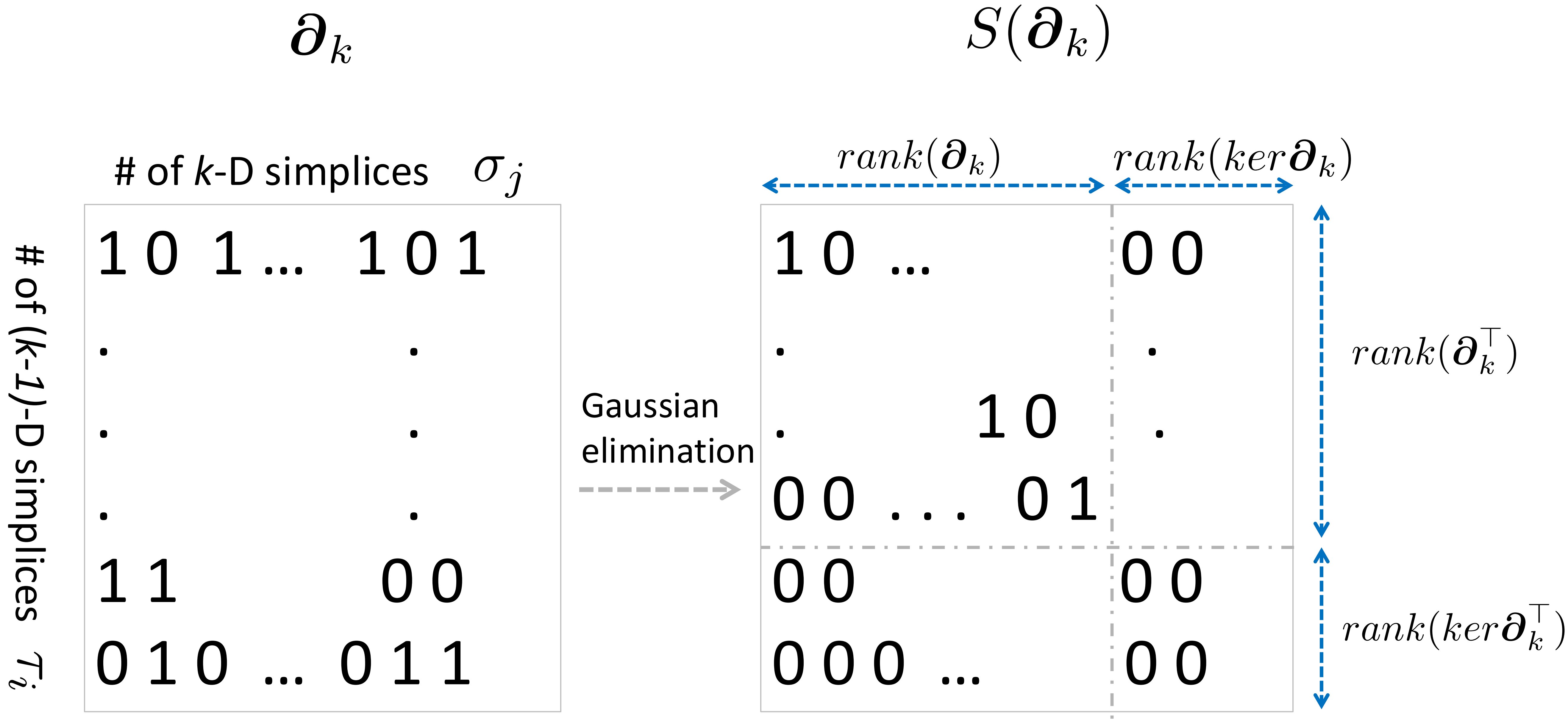
# of edges = 6

Matlab

$$
\boldsymbol{\partial}_1 = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array}
\begin{array}{c}
\begin{array}{cccccc} e_{12} & e_{23} & e_{31} & e_{24} & e_{41} & e_{45} \end{array} \\
\begin{pmatrix} -1 & 0 & 1 & 0 & 1 & 0 \\ 1 & -1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}
\end{array}
$$

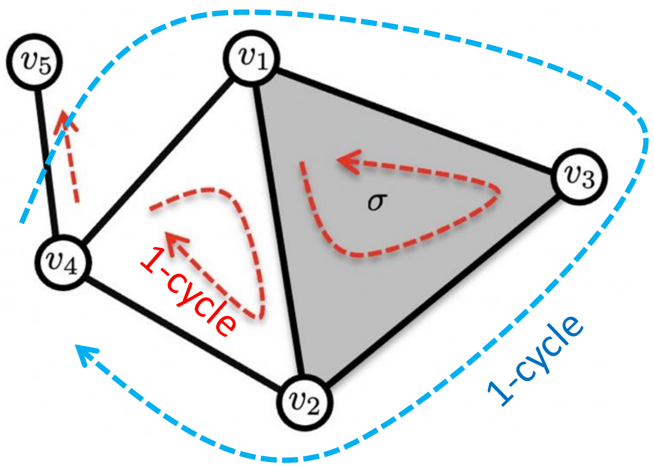
```
>M = [-1 0 1 0 1 0
      1 -1 0 -1 0 0
      0 1 -1 0 0 0
      0 0 0 1 -1 -1
      0 0 0 0 0 1]
>L=M*M'

>rank(M') = rank(L)
4

>null(M') = null(L)

   0.4472
   0.4472
   0.4472
   0.4472
   0.4472
```

Betti number computation



$$\beta_k = rank(ker\boldsymbol{\partial}_k) - rank(\boldsymbol{\partial}_{k+1})$$

Computing Betti numbers



$$\mathcal{S}(\boldsymbol{\partial}_1) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \mathcal{S}(\boldsymbol{\partial}_2) = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\beta_0 = rank(ker\boldsymbol{\partial}_0) - rank(\boldsymbol{\partial}_1)$$ = 5-4

$$\beta_1 = rank(ker\boldsymbol{\partial}_1) - rank(\boldsymbol{\partial}_2)$$ = 2-1

Far simpler approach given in *Chung et al. 2019 Network Neuroscience 3:674-694*

Hodge

(1903-1975)

Hodge theory

Hodge Laplacian

Hodge conjecture: Millennium prize problem

k-th Hodge Laplacian

$$\Delta_k = \partial_{k+1}\partial_{k+1}^\top + \partial_k^\top \partial_k$$

0th Hodge Laplacian
Graph Laplacian

# of nodes
# of nodes

$$\Delta_0 = \boxed{\partial_1 \partial_1^\top}$$

1st Hodge Laplacian

# of edges
# of edges
# of edges
# of edges

$$\Delta_1 = \boxed{\partial_2 \partial_2^\top} + \boxed{\partial_1^\top \partial_1}$$

1st Hodge Laplacian for graphs:

# of edges
# of edges

$$\Delta_1 = \boxed{\partial_1^\top \partial_1}$$

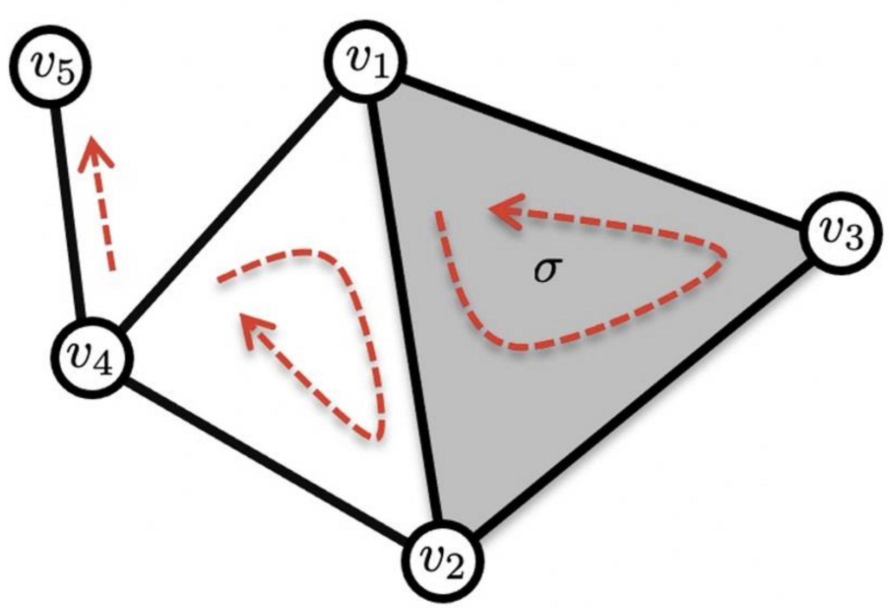
1^{st} Hodge Laplacian (for loop computation in graphs)

$$\boldsymbol{\partial}_1 = \begin{array}{c c} & \begin{array}{cccccc} e_{12} & e_{23} & e_{31} & e_{24} & e_{41} & e_{45} \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} & \begin{pmatrix} -1 & 0 & 1 & 0 & 1 & 0 \\ 1 & -1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

$$\boldsymbol{\partial}_1^\top \boldsymbol{\partial}_1 = \begin{pmatrix} 2 & -1 & -1 & -1 & -1 & 0 \\ -1 & 2 & -1 & 1 & 0 & 0 \\ -1 & -1 & 2 & 0 & 1 & 0 \\ -1 & 1 & 0 & 2 & -1 & -1 \\ -1 & 0 & 1 & -1 & 2 & 1 \\ 0 & 0 & 0 & -1 & 1 & 2 \end{pmatrix}$$

$$\boldsymbol{\partial}_2 = \begin{array}{c c} & \sigma \\ \begin{array}{c} e_{12} \\ e_{23} \\ e_{31} \\ e_{24} \\ e_{41} \\ e_{45} \end{array} & \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \end{array}$$

$$\boldsymbol{\partial}_2 \boldsymbol{\partial}_2^\top = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$



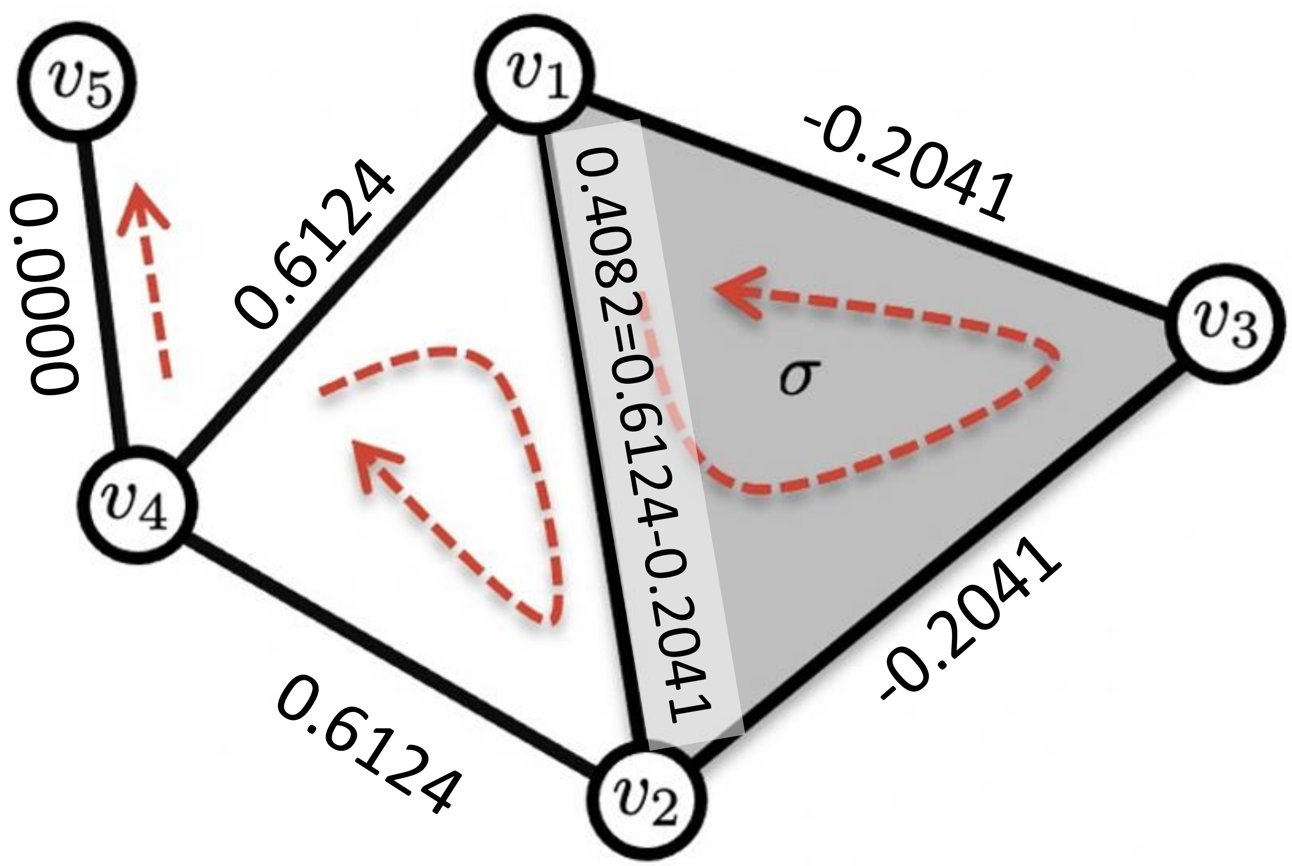
$$\Delta_1 = \begin{pmatrix} 3 & 0 & 0 & -1 & -1 & 0 \\ 0 & 3 & 0 & 1 & 0 & 0 \\ 0 & 0 & 3 & 0 & 1 & 0 \\ -1 & 1 & 0 & 2 & -1 & -1 \\ -1 & 0 & 1 & -1 & 2 & 1 \\ 0 & 0 & 0 & -1 & 1 & 2 \end{pmatrix}$$

Eigenvectors of Hodge Laplacian

$$Diag(\lambda_0, \cdots, \lambda_5) = \begin{pmatrix} 0.0000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.8299 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.6889 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3.0000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4.0000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4.4812 \end{pmatrix}$$

$$(\mathbf{f}_0, \cdots, \mathbf{f}_5) = \begin{pmatrix} 0.4082 & -0.0000 & 0.0000 & 0.5774 & 0.7071 & -0.0000 \\ -0.2041 & 0.1993 & -0.5765 & 0.5774 & -0.3536 & 0.3578 \\ -0.2041 & -0.1993 & 0.5765 & 0.5774 & -0.3536 & -0.3578 \\ 0.6124 & -0.4325 & 0.1793 & -0.0000 & -0.3536 & 0.5299 \\ 0.6124 & 0.4325 & -0.1793 & -0.0000 & -0.3536 & -0.5299 \\ 0.0000 & -0.7392 & -0.5207 & 0.0000 & 0.0000 & -0.4271 \end{pmatrix}$$

Eigenvectors of Hodge Laplacian of zero eigenvalue

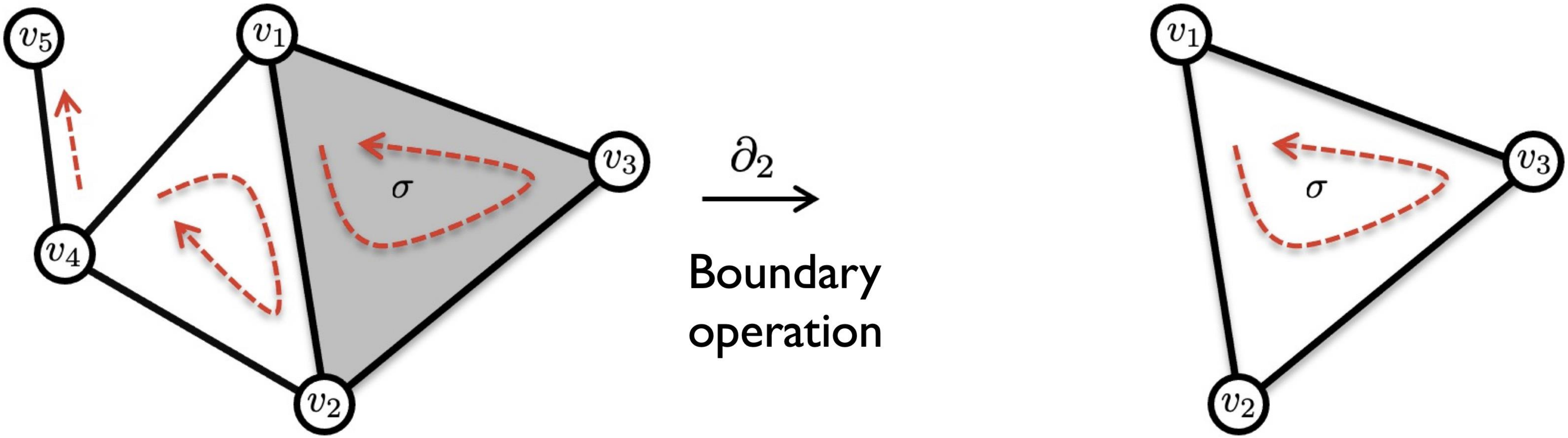


$$cycle = \sum_{i<j} a_{ij} e_{ij}$$

Method will likely only work in sparse networks without many cycles.

$$\begin{array}{cc} & \begin{array}{cccccc} a_{ij} & & & & & \end{array} \\ (\mathbf{f}_0, \cdots, \mathbf{f}_5) = & \begin{pmatrix} 0.4082 & -0.0000 & 0.0000 & 0.5774 & 0.7071 & -0.0000 \\ -0.2041 & 0.1993 & -0.5765 & 0.5774 & -0.3536 & 0.3578 \\ -0.2041 & -0.1993 & 0.5765 & 0.5774 & -0.3536 & -0.3578 \\ 0.6124 & -0.4325 & 0.1793 & -0.0000 & -0.3536 & 0.5299 \\ 0.6124 & 0.4325 & -0.1793 & -0.0000 & -0.3536 & -0.5299 \\ 0.0000 & -0.7392 & -0.5207 & 0.0000 & 0.0000 & -0.4271 \end{pmatrix} \end{array}$$

Hodge Laplacian on simplical complex vs. Graph



$$\Delta_1 = \boldsymbol{\partial}_2 \boldsymbol{\partial}_2^\top + \boldsymbol{\partial}_1^\top \boldsymbol{\partial}_1$$

$$\Delta_1 = \boldsymbol{\partial}_1^\top \boldsymbol{\partial}_1$$

If we form 2-simplx for every 3 nodes, we cannot have a cycle with 3 nodes.

If we treat all three nodes to form a cycle, there will be too many cycles

vector representation of 1-cycles

$$cycle = \sum_{i<j} a_{ij} e_{ij}$$

$$\begin{array}{cc} & \beta_1 \\ \text{\# of edges} & \begin{array}{|ccc|} \hline 2.3 & \dots & 1.39 \\ . & & . \\ . & & . \\ . & & . \\ -1.4 & \dots & 0 \\ \hline \end{array} \end{array}$$

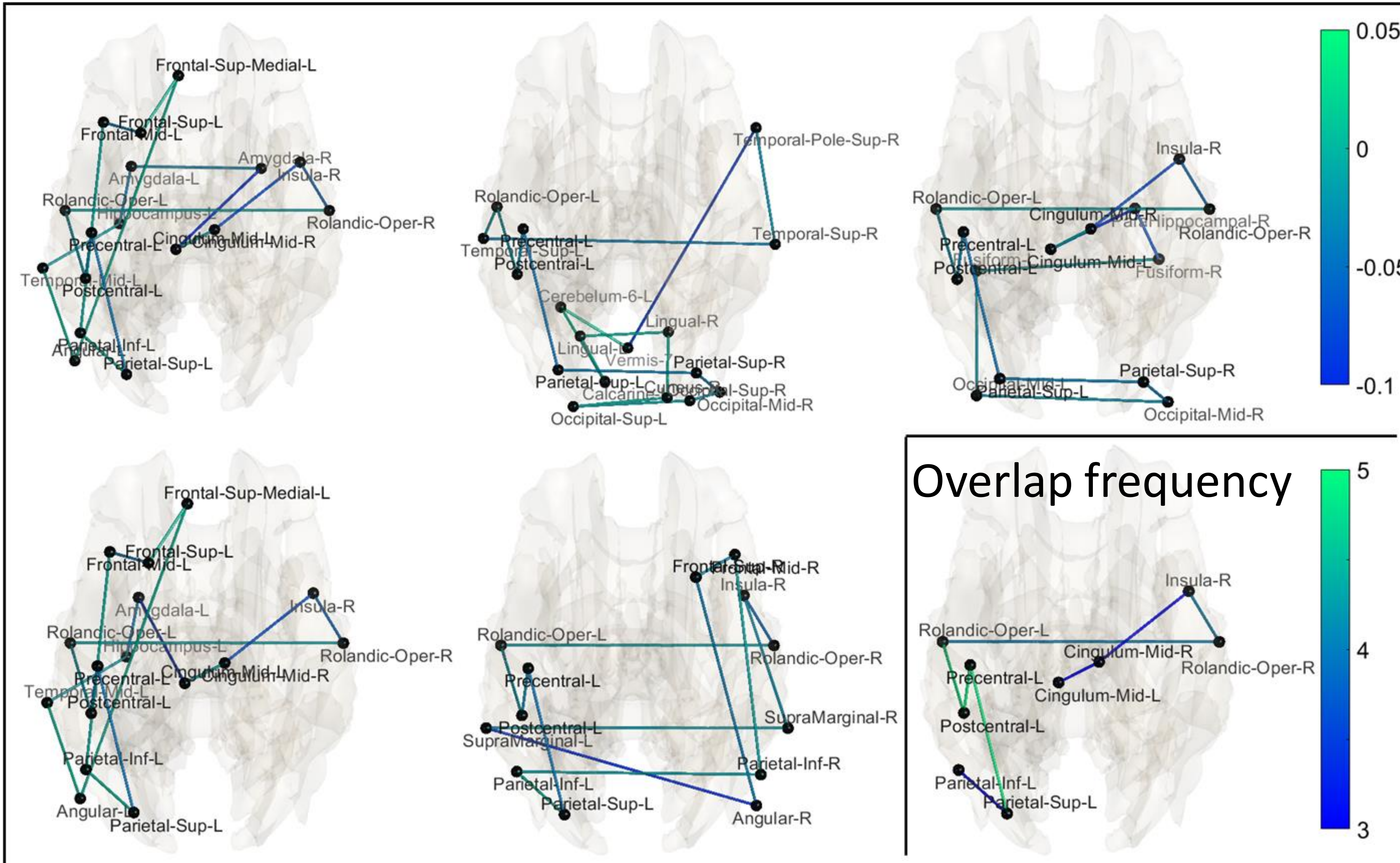
Linearly independent columns

$\xrightarrow{\text{Gaussian elimination}}$

$$\begin{array}{cc} & \beta_1 \\ \text{\# of edges} & \begin{array}{|ccccc|} \hline 1 & 0\dots & & & 0 \\ 1 & 0 & & . & 0 \\ 0 & 1 & & . & 0 \\ 0 & 1 & & . & 1 \\ 1. & \dots & 0 & & 1 \\ \hline \end{array} \end{array}$$

More meaningful representation

Fourier analysis on cycles → Five dominating cyclic basis



Lee et al. 2014 MICCAI 8675:297-304

<u>Anand and Chung 2023, IEEE TMI</u> Hodge Laplacian of Brain

cycles across brain networks?

Lee et al. 2019 MICCAI 11767:674-682

http://pages.stat.wisc.edu/~mchung/papers/lee.2019.MICCAI.pdf

2.3 Stiefel Optimization for Group-Level Harmonic Forms

Given a network K, the problem of estimating harmonic k-forms of $\boldsymbol{L}_k$ can be written by an optimization problem on a Stiefel manifold

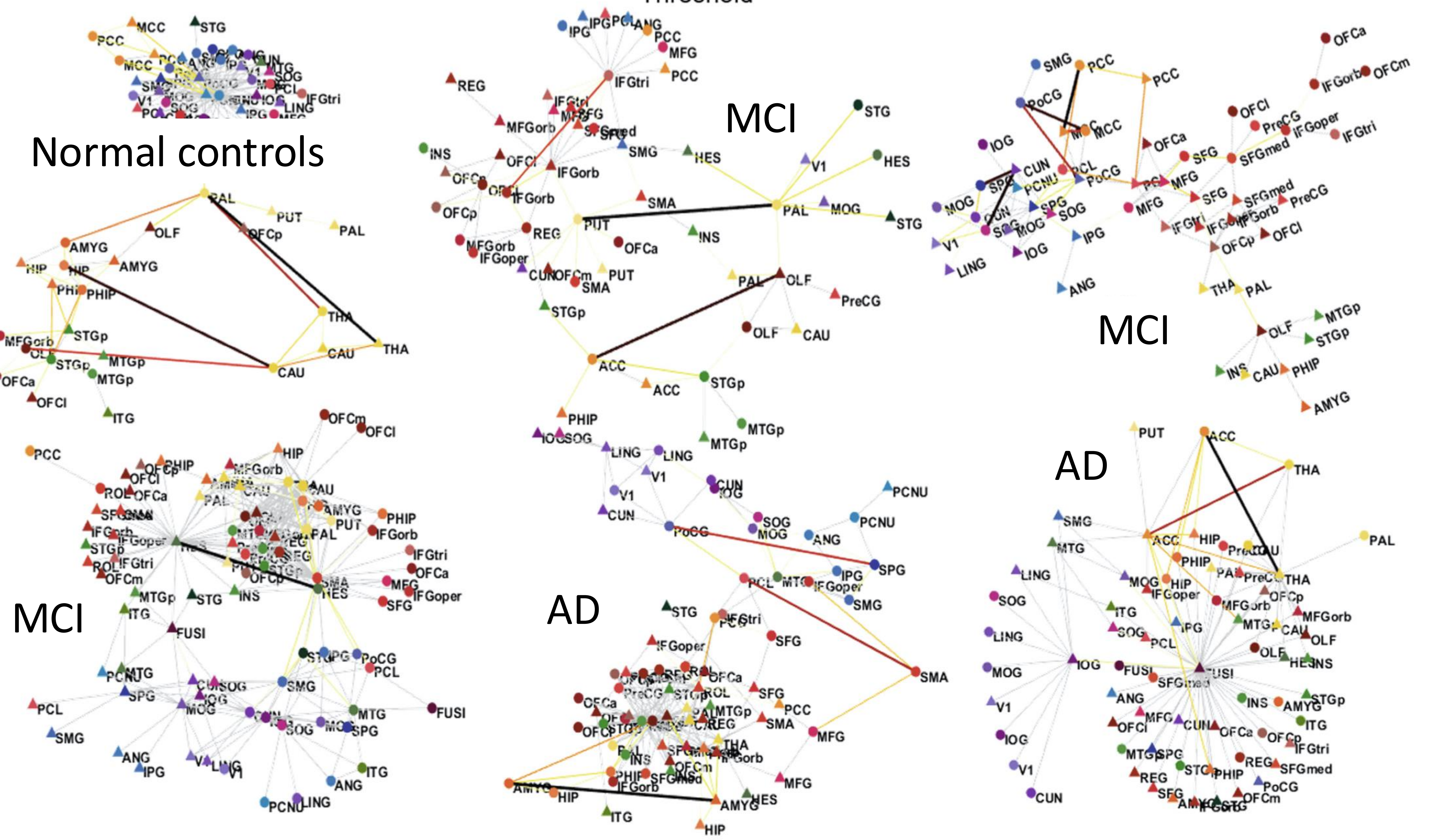
$$\min_{H_k \in \mathcal{S}(|K_k|, r)} \operatorname{tr} \boldsymbol{H}_k^\top \boldsymbol{L}_k \boldsymbol{H}_k + \beta \parallel \boldsymbol{H}_k \parallel_1, \tag{3}$$

where $\mathcal{S}(|K_k|, r)$ is a Stiefel manifold which is the set of all r-tuples of orthonormal vectors in $\mathbb{R}^{|K_k|}$, $\| \cdot \|_1$ is the l_1-norm of $\cdot$, and β is the control parameter for sparseness [8].

Pairwise Similarity Constraint for Group Analysis. Suppose that there are N simplicial networks in a group. Their kth Hodge Laplacians and harmonic forms are denoted by $\boldsymbol{L}_k^{(1)}, \ldots, \boldsymbol{L}_k^{(N)}$ and $\boldsymbol{H}_k^{(1)}, \ldots, \boldsymbol{H}_k^{(N)}$, respectively. To estimate group-level harmonic forms, we extend (3) to

$$\min_{H_n \in \mathcal{S}(|K_k|, r)} \sum_{n=1}^{N} \operatorname{tr} \Big((\boldsymbol{H}_k^{(n)})^\top \boldsymbol{L}_k^{(n)} \boldsymbol{H}_k^{(n)} + \beta \parallel \boldsymbol{H}_k^{(n)} \parallel_1 \Big)$$

The probability of having identical joint cycles is very low



Difficult biological interpretation

Hodge Decomposition

HODGE-DECOMPOSITION OF BRAIN NETWORKS

D Vijay Anand[1], *Moo K. Chung*[2]

[1] University College London, London, UK
[2] University of Wisconsin, Madison, USA
ucahvad@ucl.ac.uk, mkchung@wisc.edu

arXiv:2211.10542

Hodge decomposition (orthogonal)

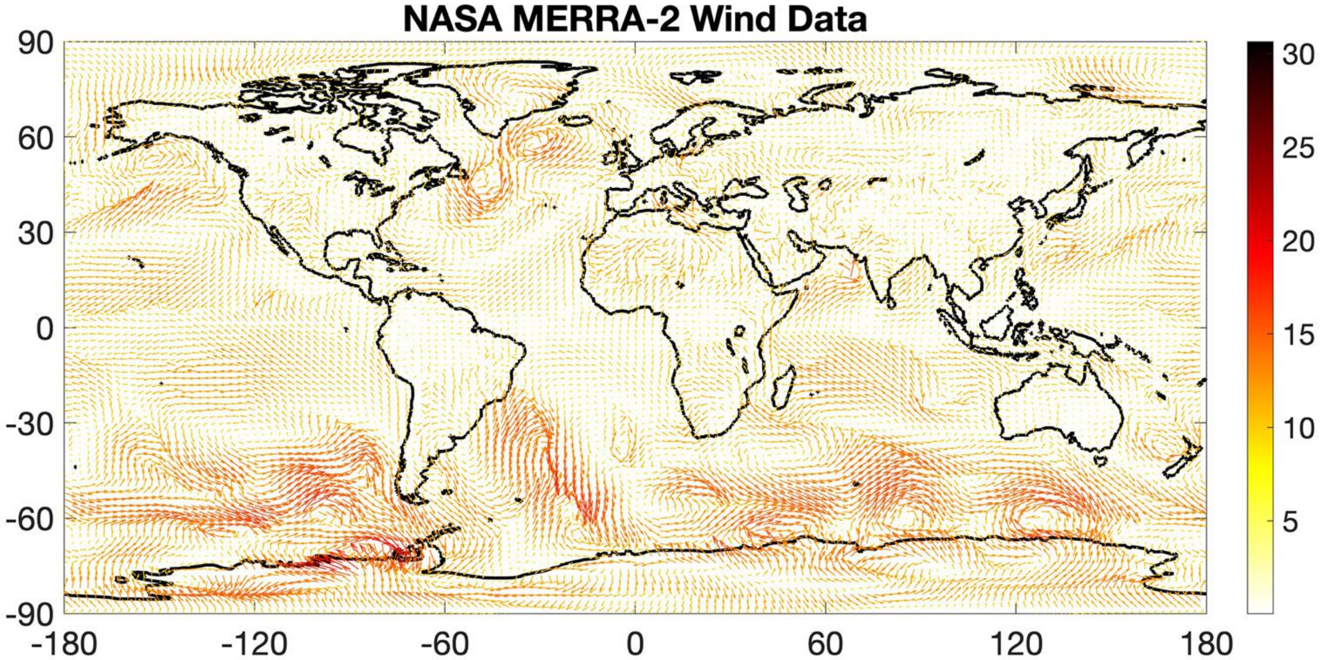
Edge flow | Gradient | Curl | Harmonic

$$X = X_G + X_C + X_H = \boxed{\partial_1^\top s} + \boxed{\partial_2 \phi} + X_H$$

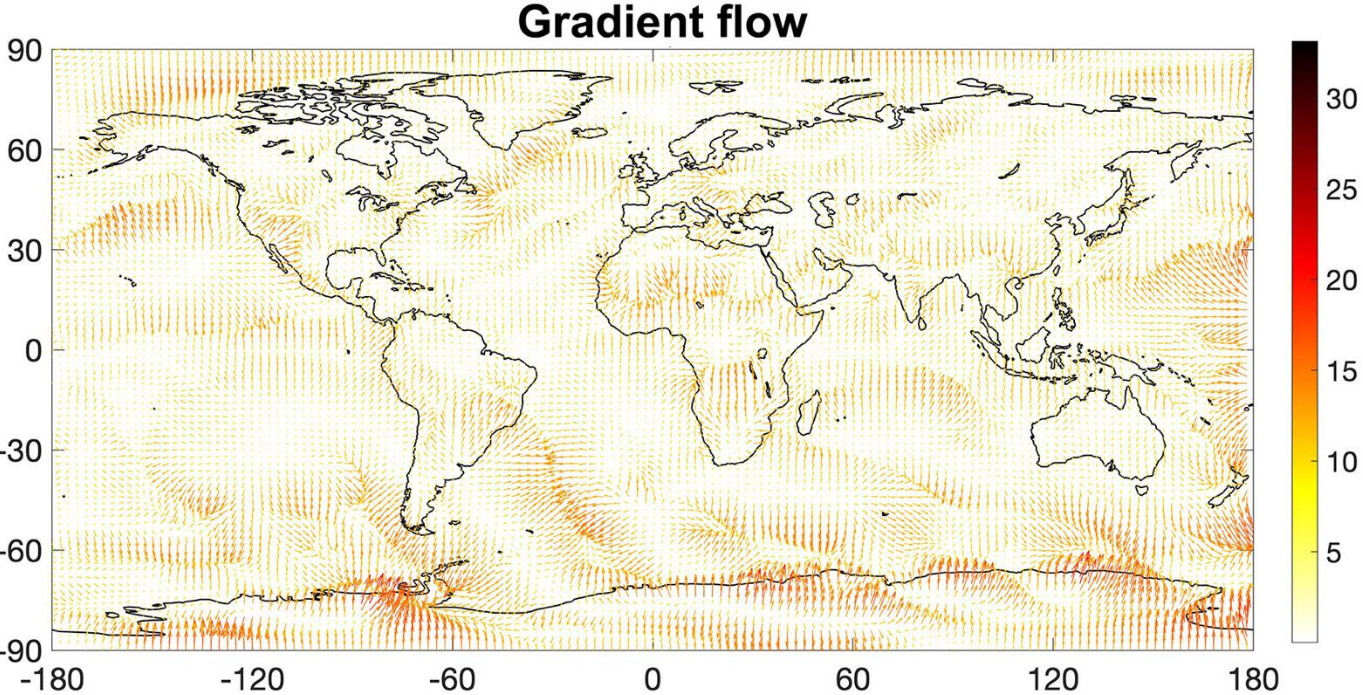
Function over edge

Function over triangle

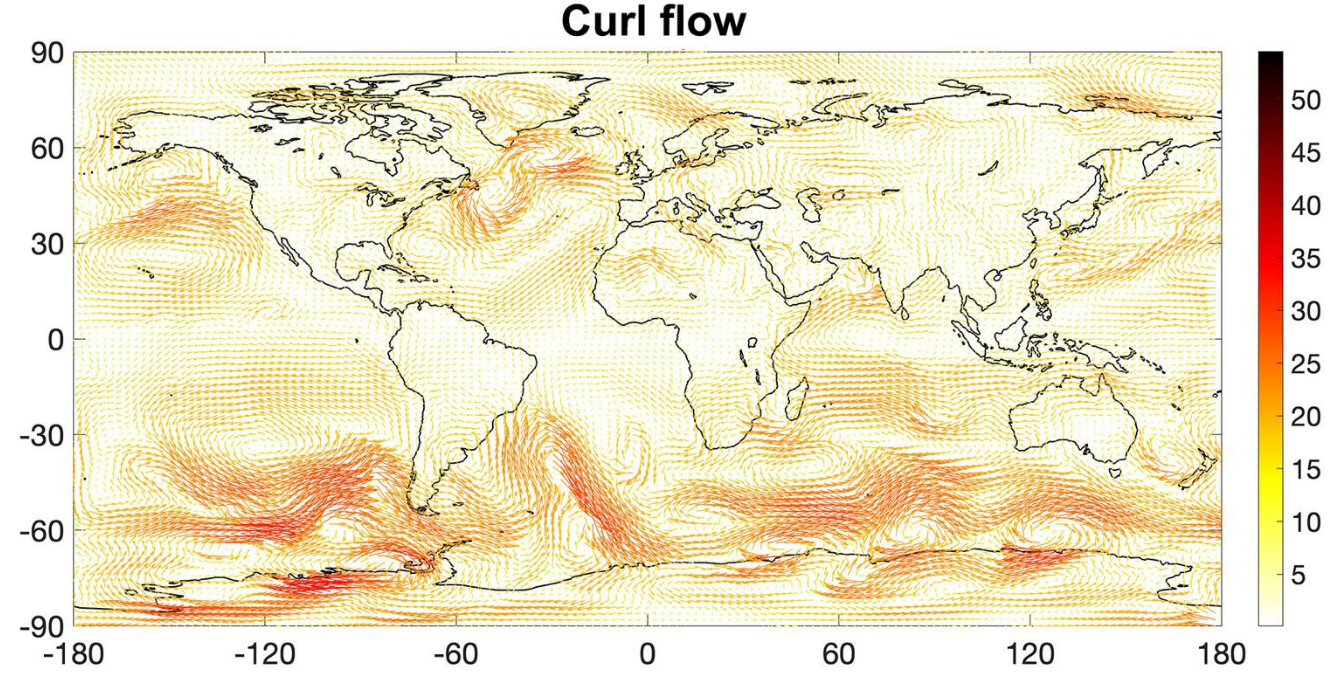
NASA MERRA-2 Wind Data



Gradient flow



Curl flow



Hodge decomposition

Edge flow

Gradient

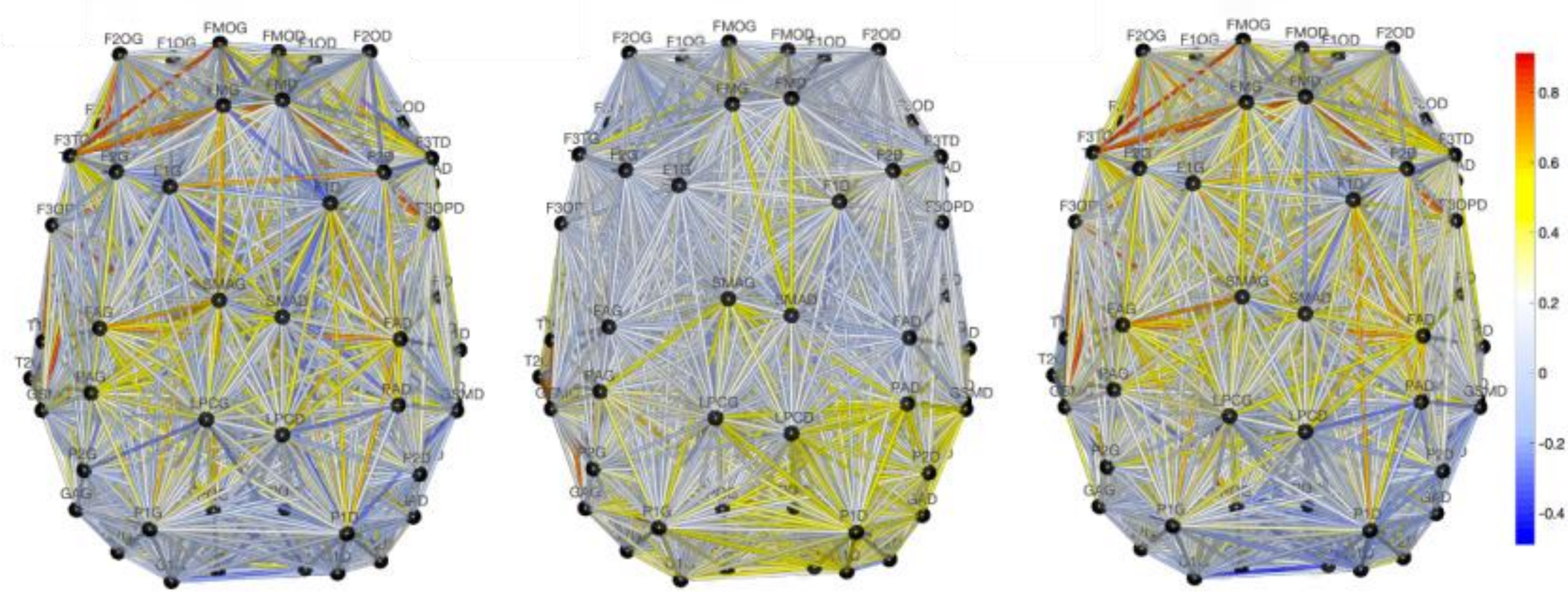
Curl

Harmonic

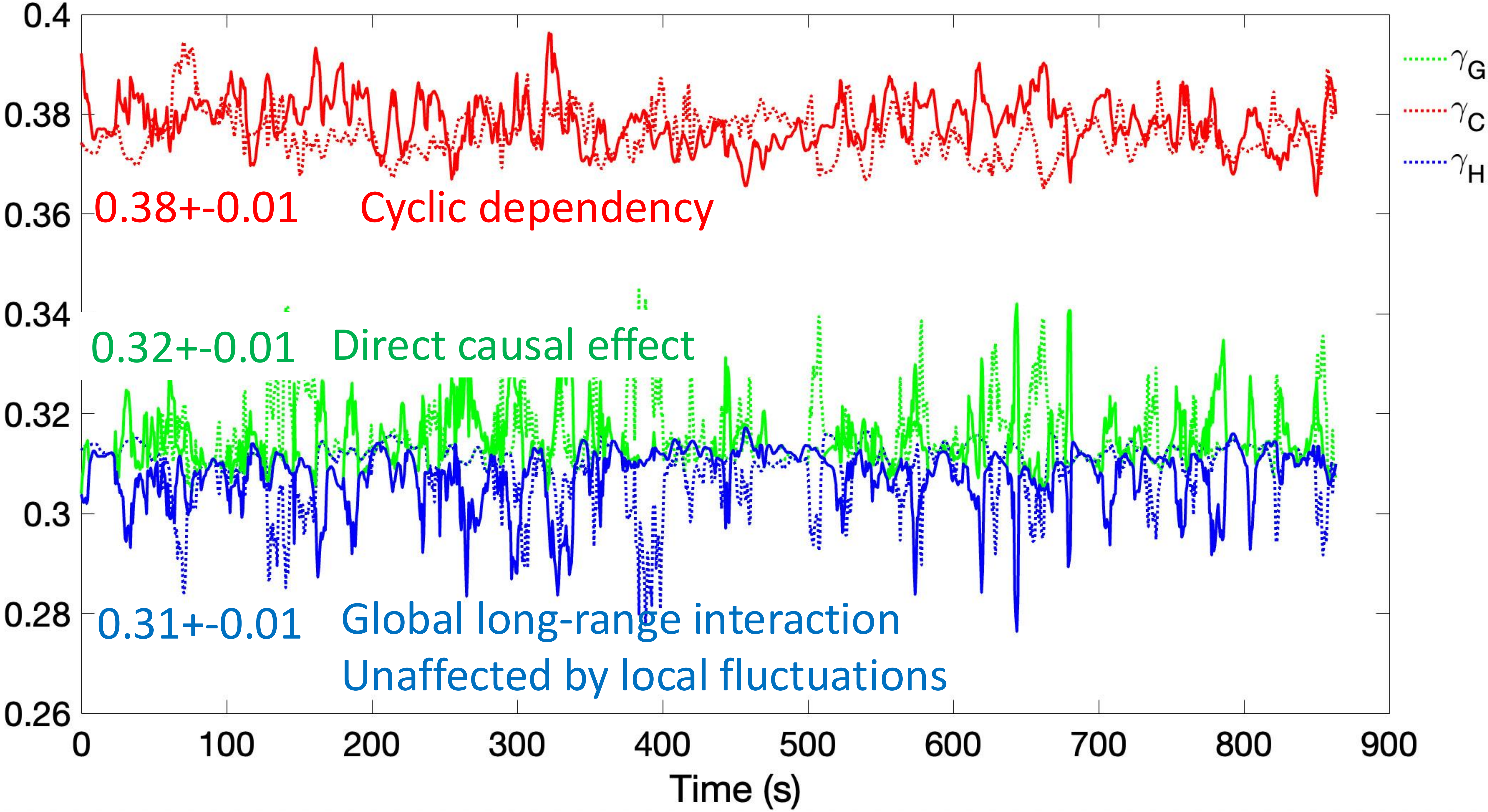
$$X = X_G + X_C + X_H = \boxed{\partial_1^\top s} + \boxed{\partial_2 \phi} + X_H$$

Function over edge

Function over face



Dynamics of Hodge decomposition of brain network

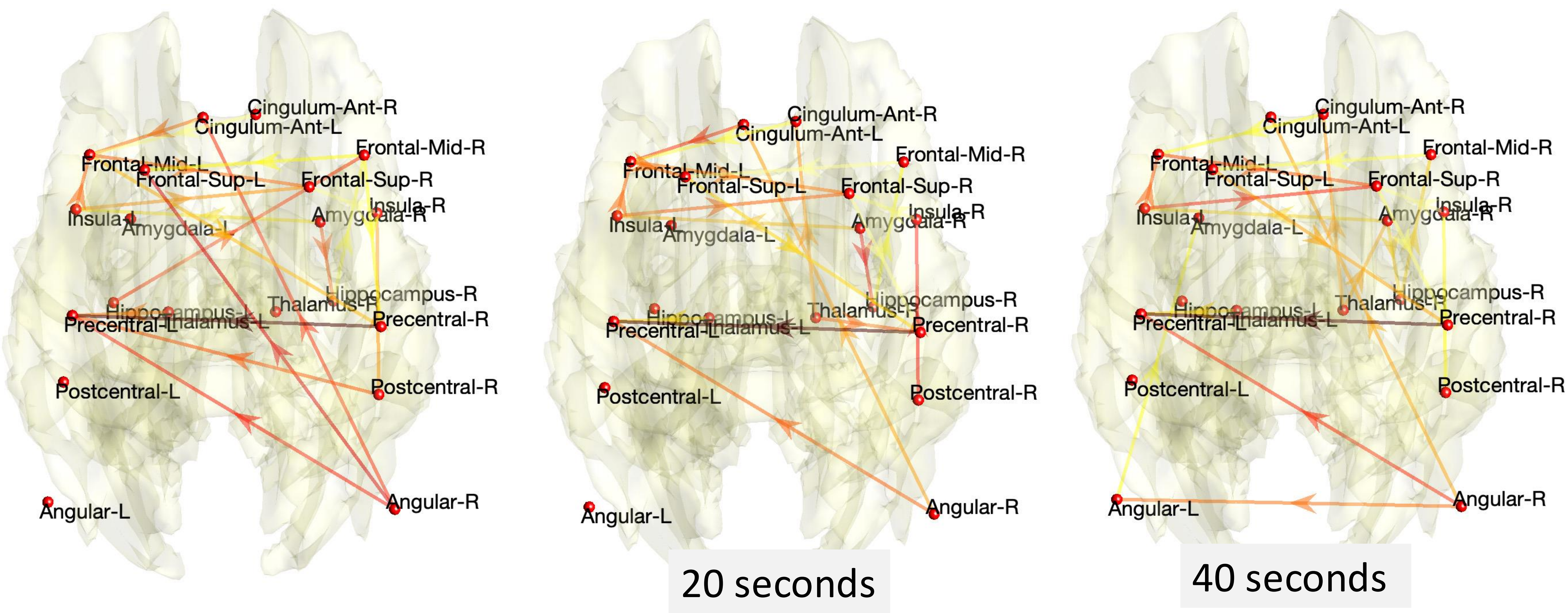


$$\gamma_G = \frac{\|\boldsymbol{\partial}_1^\top s\|_2}{\|vec\ X\|_2}, \gamma_C = \frac{\|\boldsymbol{\partial}_2 \phi\|_2}{\|vec\ X\|_2}, \gamma_H = 1 - \gamma_G - \gamma_C$$

Topological Causal Model of dynamic brain network

Stability of direct causal paths over 40 seconds

20 largest gradients components



Right precentral gyrus to the left precentral gyrus has the highest causal ranking (largest magnitude of gradient)