Control Principles for Engineered Systems 5SMC0
Control principles for linear systems 5ARD0

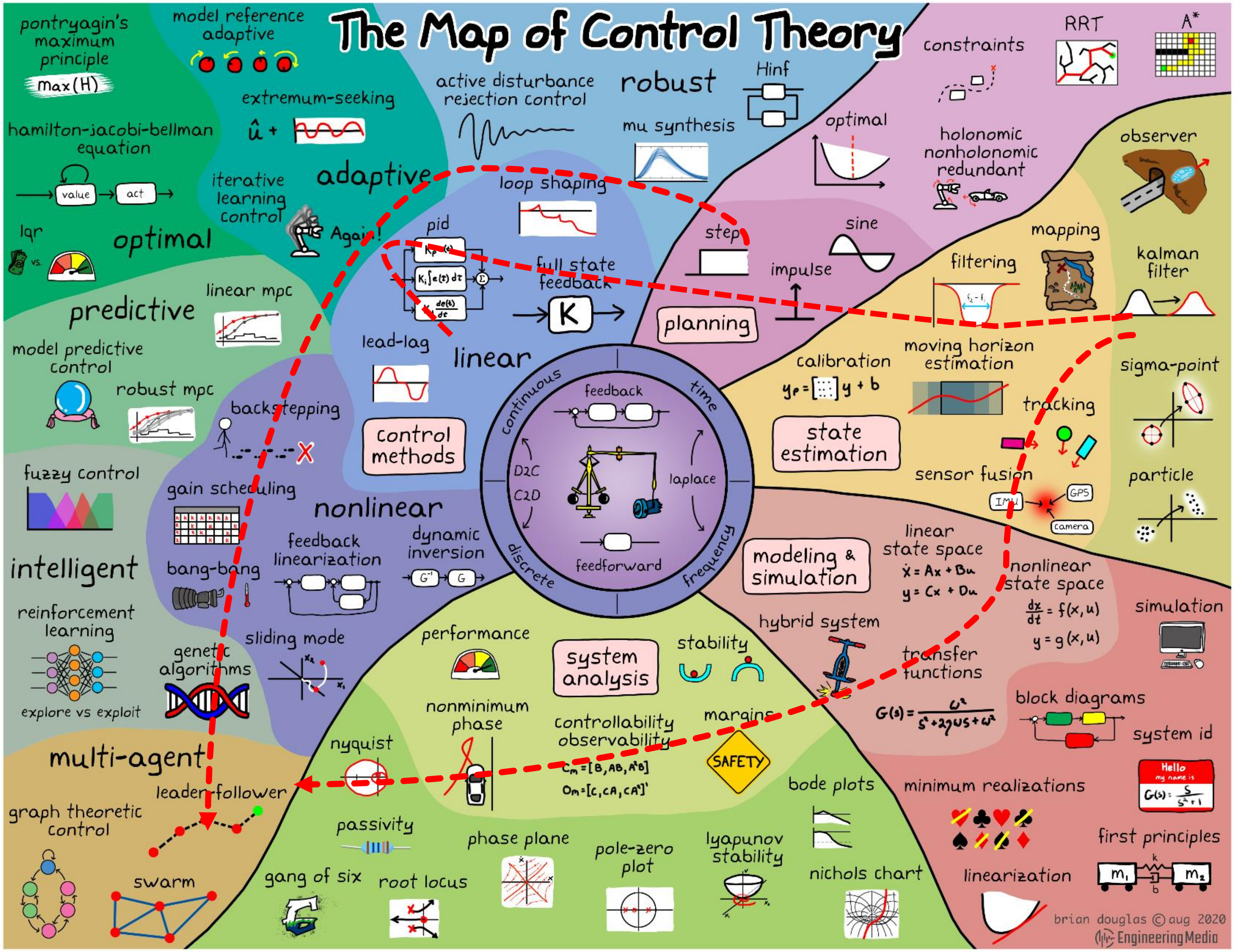
Networked Control System

Lecture 1

Zhiyong Sun

Control Systems Group
Department of Electrical Engineering
Eindhoven University of Technology

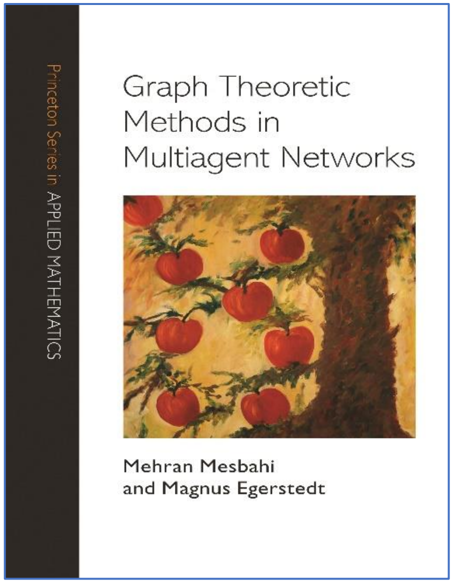
Where are we now (in the map of control theory)



Networked Control System

Lecture 1

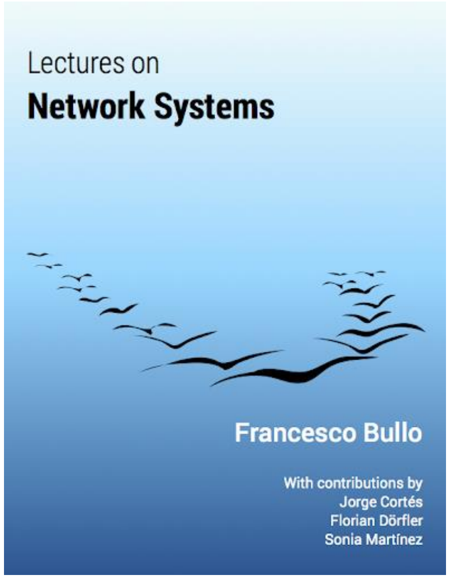
Introduction to Network System & Graph Theory


Princeton Series in APPLIED MATHEMATICS

Graph Theoretic
Methods in
Multiagent Networks

Mehran Mesbahi
and Magnus Egerstedt

Chapter 2


Lectures on
Network Systems

Francesco Bullo

With contributions by
Jorge Cortés
Florian Dörfler
Sonia Martínez

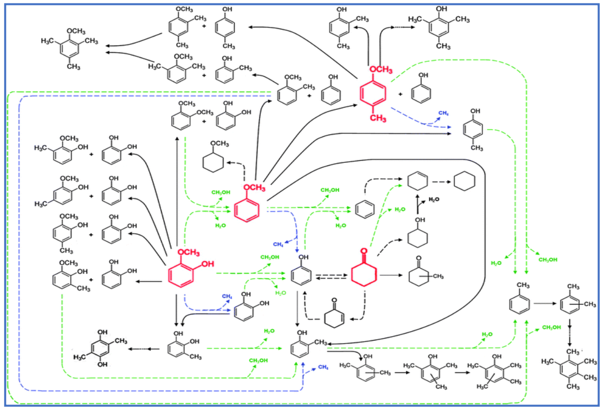
Chapters 3-4

Outline

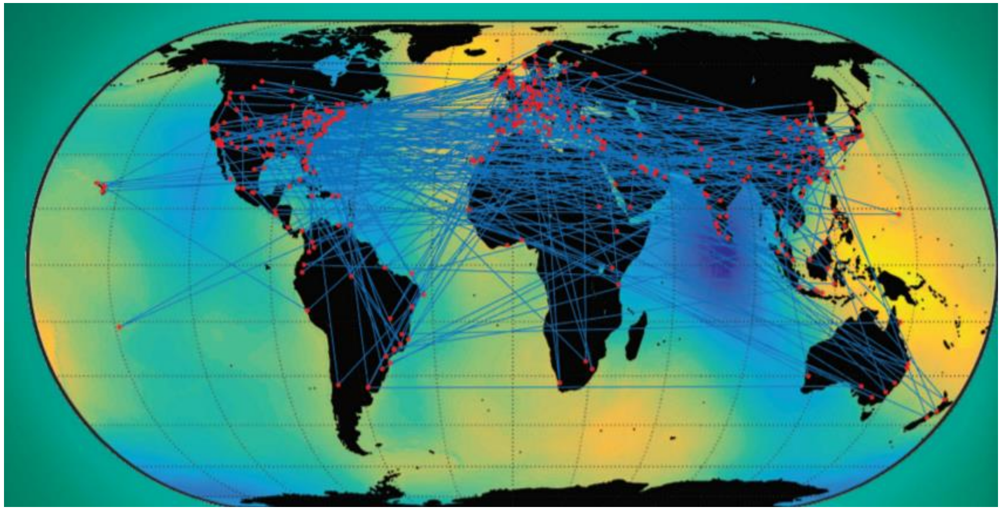
- Modelling of network systems
- Basic graph theory
- Graph and matrices
- Multi-agent consensus

Complex network systems

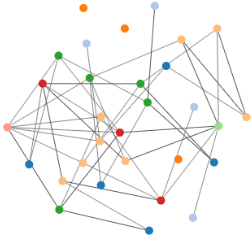
Sensor networks



Chemical reaction network

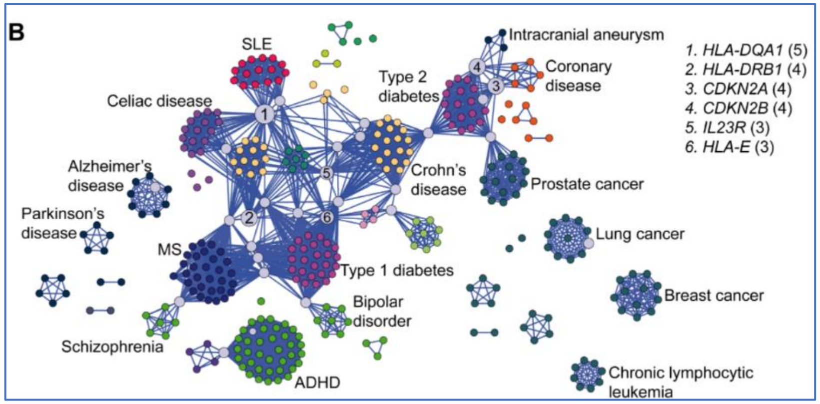


Airport traffic network (IEEE CSM 2016)



Complex human disease gene network

(Barrenas et al. Plos One 2009)



Emerging behavior in network systems

Bird flocking

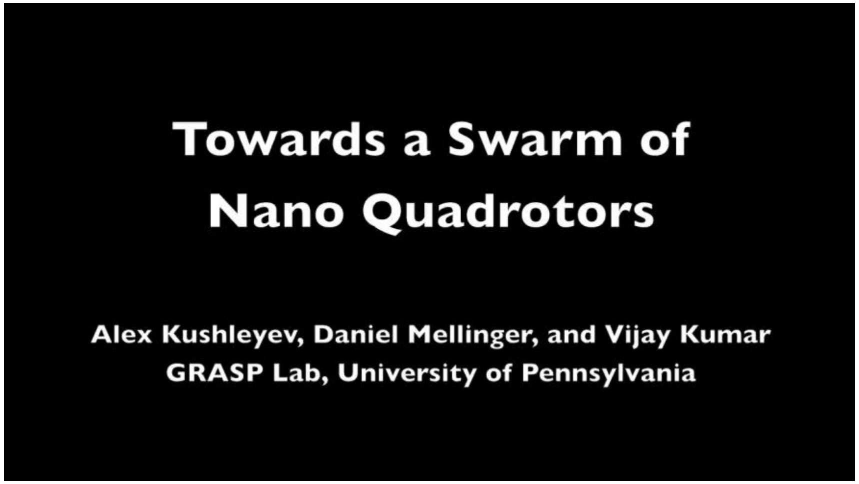
Fish schooling

Synchronization/consensus (@UCLA)

(**Coupling leads to synchronization)**

https://www.youtube.com/watch?v=T58lGKREubo

A swarm of nano quadrotors (@U Pen)



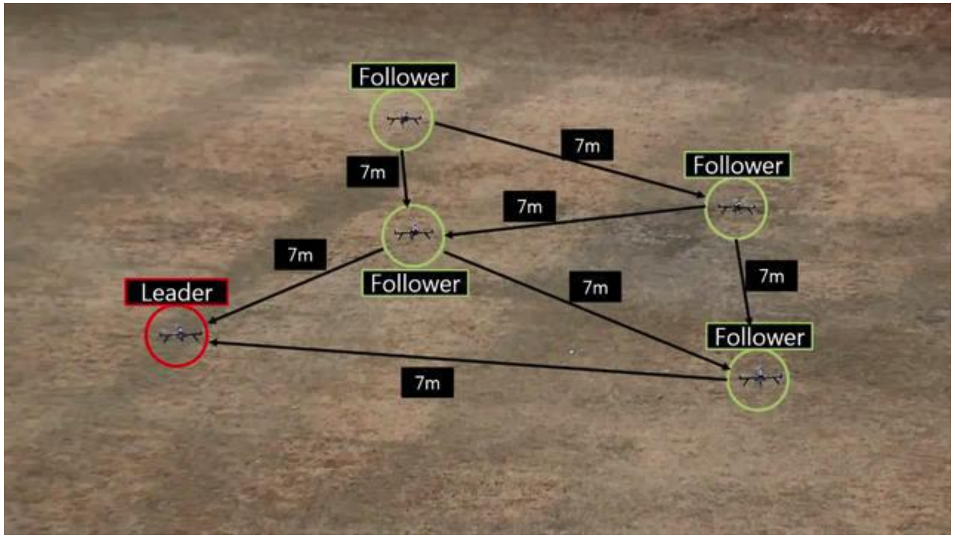
Towards a Swarm of
Nano Quadrotors

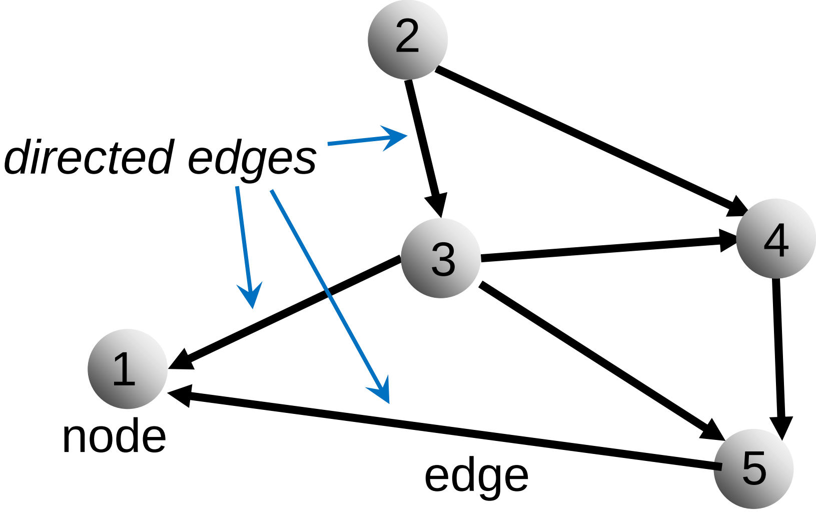
Alex Kushleyev, Daniel Mellinger, and Vijay Kumar
GRASP Lab, University of Pennsylvania

(Relative sensing/communication for swarm & coordination)

https://www.youtube.com/watch?v=YQIMGV5vtd4

Abstraction of a network system (in graph modelling)





A multi-drone formation ⇒ A graph

Network systems can be represented by graphs

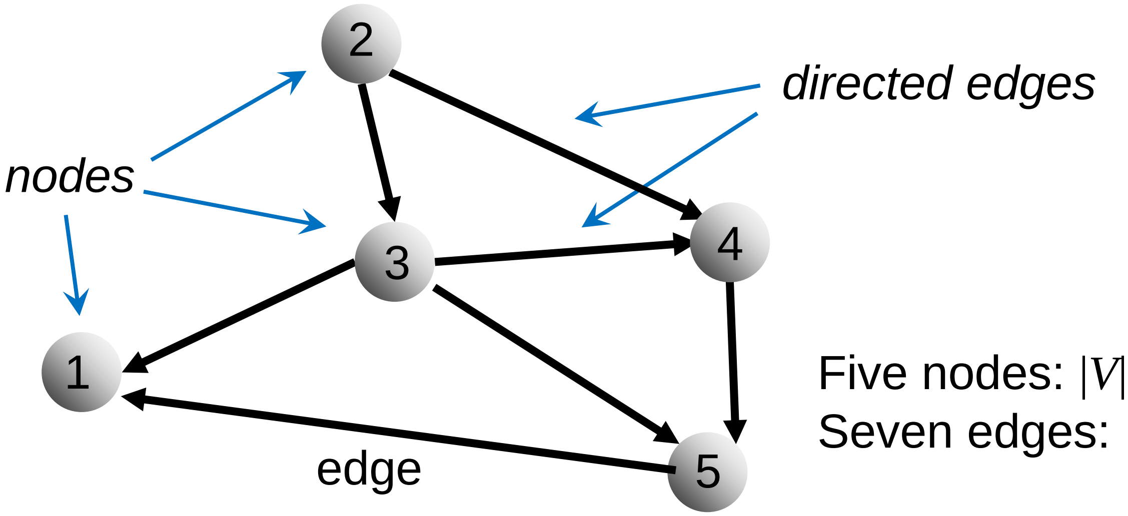
- *Drones/vehicles are nodes (vertices)*
- *Communication/sensing links are edges*

A node has dynamics and states, e.g., the position, attitude, velocity, rotation, etc.

Communication/sensing links describe the interaction of relative states (e.g., relative positions, distances etc.)

Basic graph theory

Graph modelling: directed graph

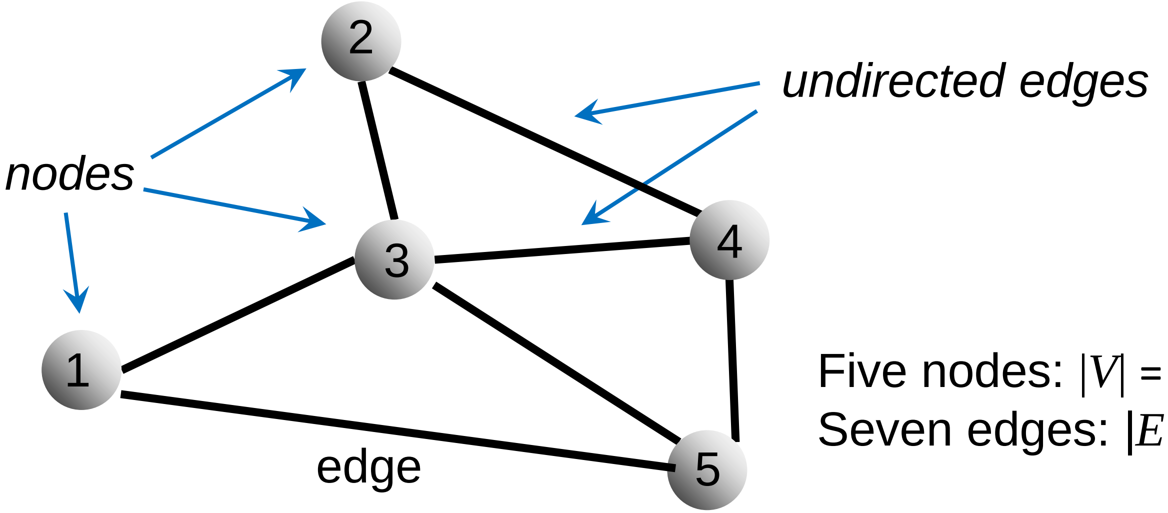


Five nodes: $|V| = 5$
Seven edges: $|E| = 7$

- An edge incident from *i* to *j* is an ordered pair (*i*, *j*)
  - i is the in-neighbor of *j*, and *j* is the out-neighbor of i
  - $\mathcal{N}_i^-$ ($\mathcal{N}_i^+$) : set of all the in-neighbors (out-neighbors) of i
  - *Example:* $\mathcal{N}_4^- = \{2, 3\}$, $\mathcal{N}_3^+ = \{1, 4, 5\}$

- A directed graph *G*:= (*V*, *E*) is two sets of objects
  - *Node set* V = { 1, 2, 3, 4, 5 }
  - *Edge set* E = { (2,3), (2,4), (3,1), (3,4), (3,5), (4,5), (5,1)}

Graph modelling: Undirected graph



Five nodes: $|V| = 5$
Seven edges: $|E| = 7$

- An edge between i to j is an unordered pair $(i, j) = (j, i)$
  - i is the neighbor of j, and vice versa
  - $\mathcal{N}_i$: the set of all the neighbors of i, *e.g.,* $\mathcal{N}_3 = \{1, 2, 4, 5\}$

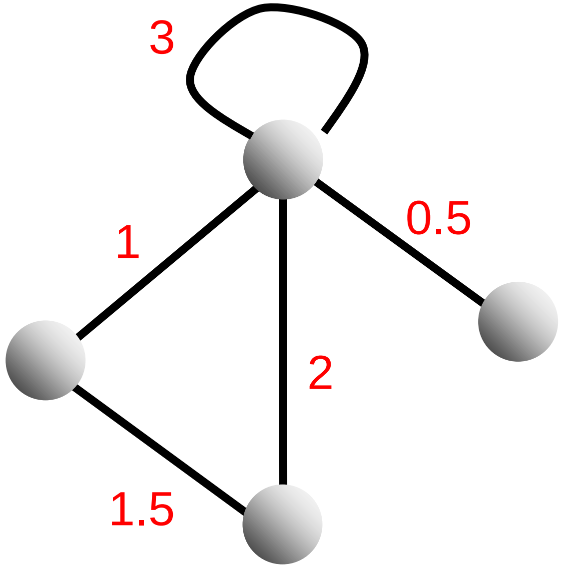
- An undirected graph $G := (V, E)$ is two sets of objects
  - *Node set* $V = \{ 1, 2, 3, 4, 5 \}$
  - *Edge set* $E = \{ (2,3), (2,4), (3,1), (3,4), (3,5), (4,5), (5,1) \}$

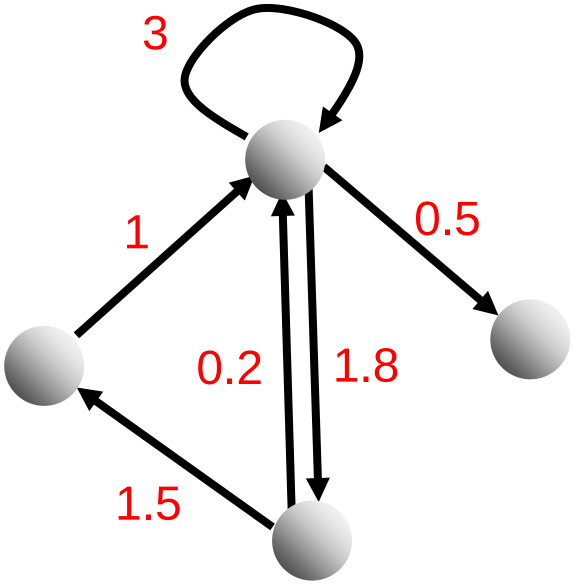
    or $E = \{ (3,2), (4,2), (1,3), (4,3), (5,3), (5,4), (1,5) \}$

(undirected edges, the orders do not matter)

Weighted graph

- A (directed/undirected) graph is *weighted* if each edge has an associated weight.

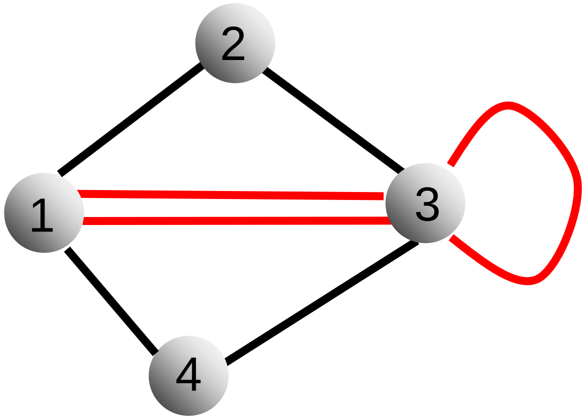


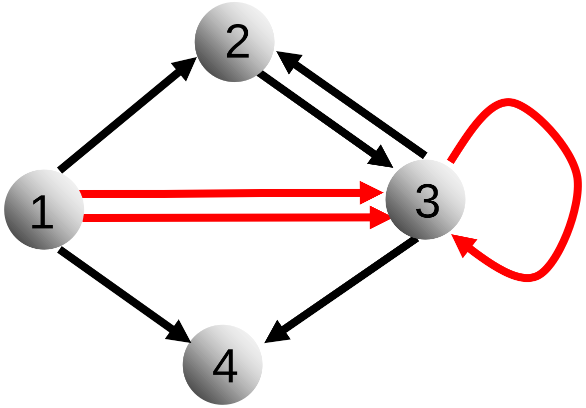


- The weight can be positive, negative (e.g., signed graphs), or zero (zero indicates the absence of connections).
  - In this course we consider positive weights.

Simple graphs

- Parallel edges
  - Two or more edges linking the same (ordered) pair of nodes
    - e.g., *from node 1 to node 3, there are two parallel edges*
- Self-loops
  - An edge that starts and ends at the same node
    - e.g., *node 3 has a self-loop*

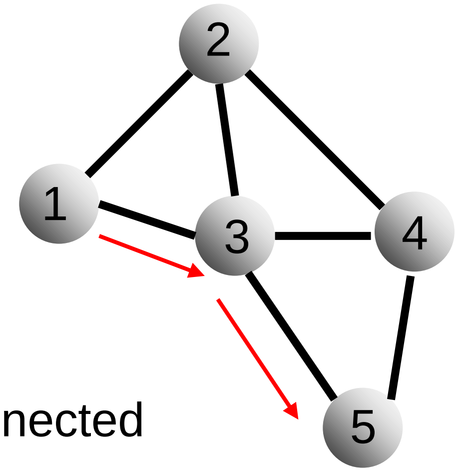




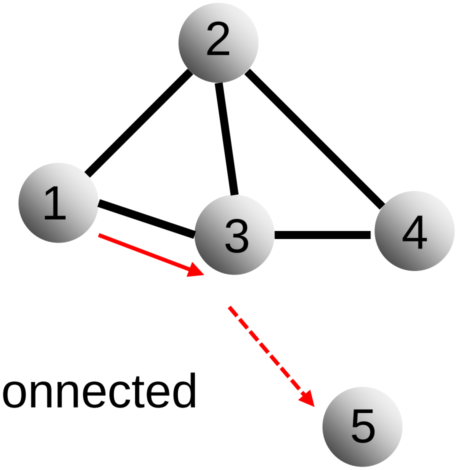
- A graph is *simple* if it has no parallel edges and self-loops.
  - In this course we consider simple graphs.

Connectivity in undirected graphs

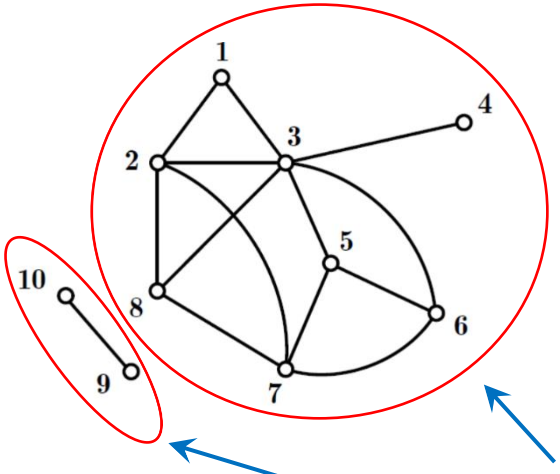
- An undirected graph $G := (V, E)$ is connected, if there is a path from i to j, for all i, j in V.



connected



not connected



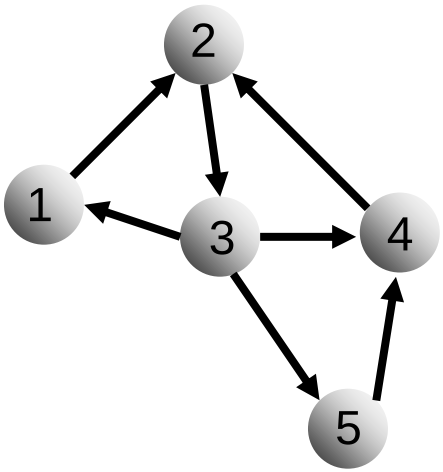
$$V = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$$

$$E = \{(1{,}2), (1{,}3), (2{,}2), (2{,}3), (2{,}7), (3{,}4), (3{,}5), (3{,}6), (3{,}8), (5{,}6), (5{,}7), (6{,}7), (7{,}8), (9{,}10)\}$$

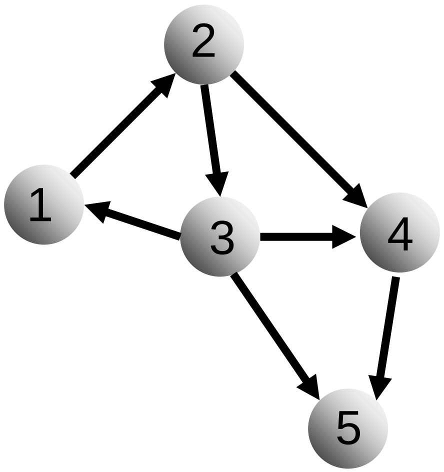
Connected components

Connectivity in directed graphs

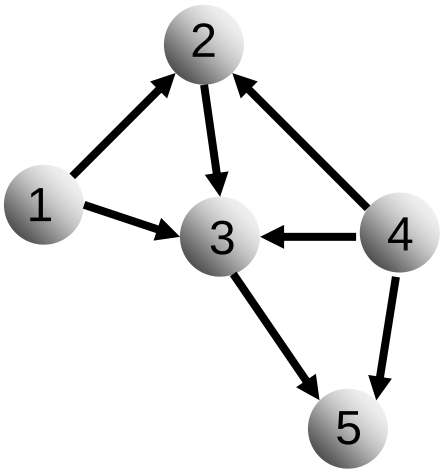
- A directed graph is
  - strongly connected, if each node can reach all the other nodes via directed paths.
  - quasi-strongly connected, if there is a node (called root node) that can reach all the other nodes via directed paths.
  - weakly connected, if the underlying undirected graph is connected.



Strongly connected



Quasi-strongly connected

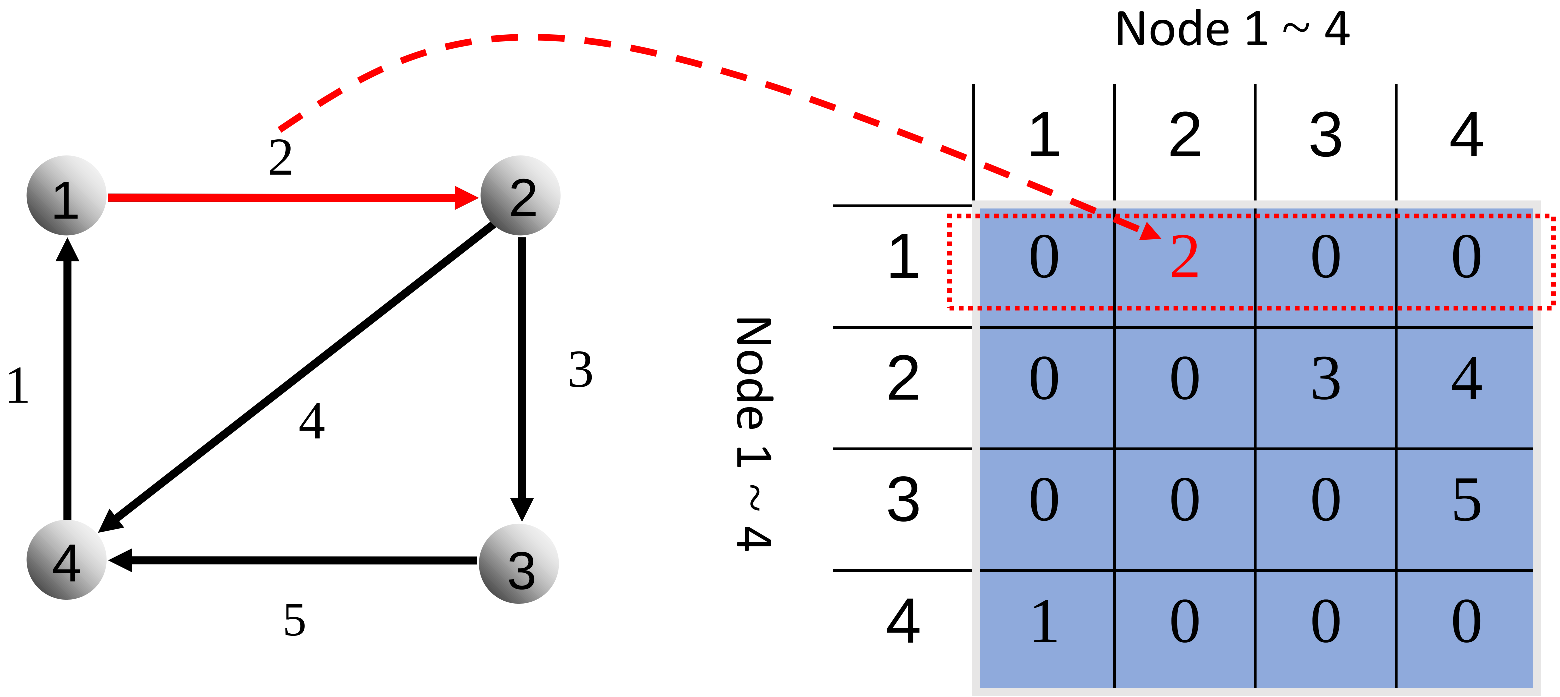


Weakly connected

Matrix representation for graphs

Adjacency matrix for *directed* graph

- *Adjacency matrix*: *A*
  - Rows and columns are labeled with ordered vertices
  - A_{ij}= 0 if no edge exists between nodes *i* and *j*
  - A_{ij} is the weight of the edge incident from *i* to *j*



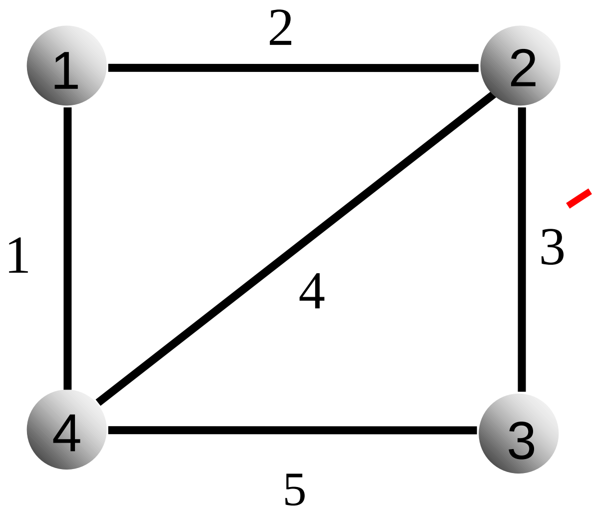
	1	2	3	4
1	0	2	0	0
2	0	0	3	4
3	0	0	0	5
4	1	0	0	0

Directed graph *G*

A is square and asymmetric

Adjacency matrix for *undirected* graph

- *Adjacency matrix*: *A*
  - Rows and columns are labeled with ordered vertices
  - A_{ij}= 0 if no edge exists between nodes *i* and *j*
  - A_{ij} is the weight of the edge (*i*, *j*) connecting nodes *i* and *j*



Undirected graph *G*

Node 1 ~ 4

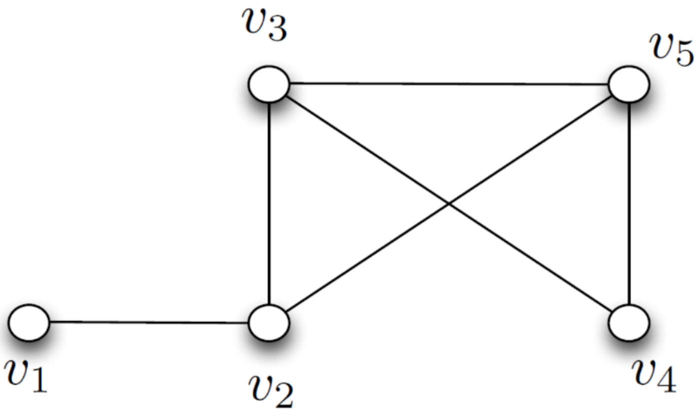
Node 1 ~ 4

	1	2	3	4
1	0	2	0	1
2	2	0	3	4
3	0	3	0	5
4	1	4	5	0

A is square and symmetric!

Adjacency/degree matrix

- *Example: adjacency matrix A*



$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

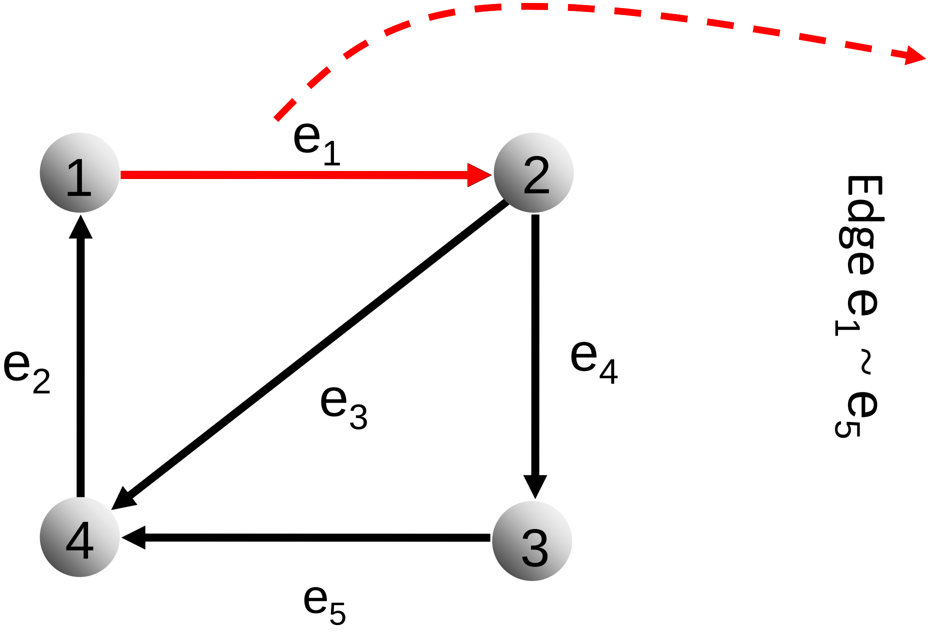
- *Example: diagonal degree matrix D*
  - D_{ii} : sum of the (weighted out-) degrees of the edges from node i

$$D = diag\left(\sum_{j=1}^{N} A_{ij}\right)$$

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

Incidence matrix for *directed* graph

- *Incidence matrix*: *H*
  - Label rows with edges and columns with vertices (nodes)
  - H_{ij}= 1 if edge e_i is incident from node *j*
  - H_{ij}= -1 if edge e_i is incident into node *j*
  - H_{ij}= 0 otherwise



Node 1 ~ 4

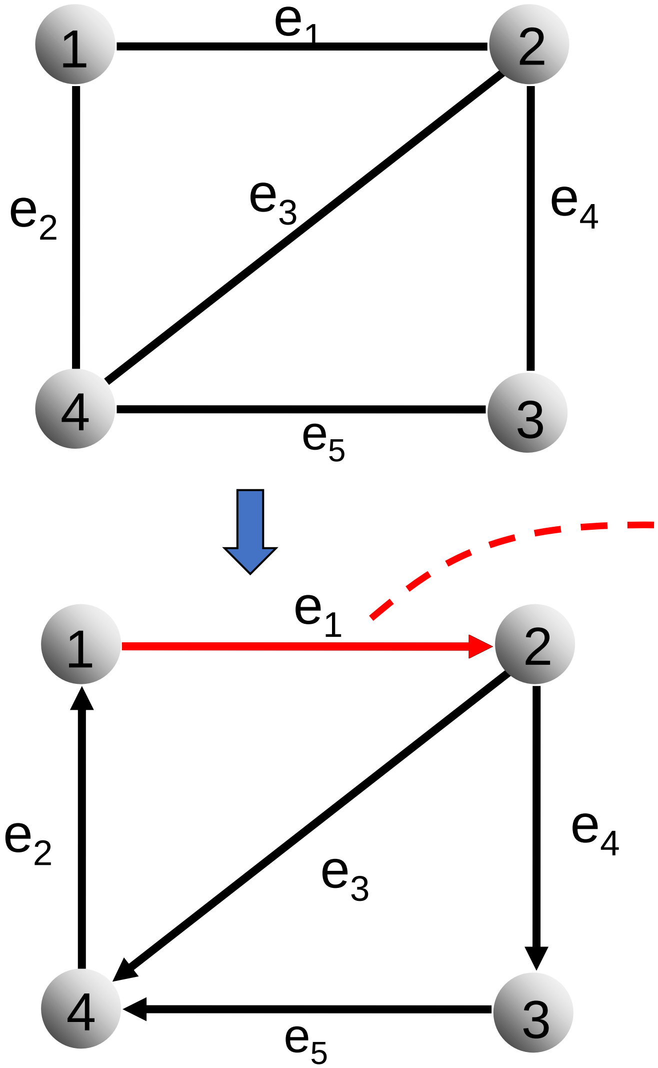
Edge $e_1 \sim e_5$

	1	2	3	4
e_1	1	-1	0	0
e_2	-1	0	0	1
e_3	0	1	0	-1
e_4	0	1	-1	0
e_5	0	0	1	-1

➢ Note: The other convention to define incidence matrix is to label rows with vertices, and columns with edges (i.e., the transpose of *H*).

Incidence matrix for *undirected* graph

- For an undirected graph, the incidence matrix is obtained by assigning an arbitrary direction to each edge



- *Incidence matrix*: *H*
  - H_{ij}= 1 if edge e_i is incident from node *j*
  - H_{ij}= -1 if edge e_i is incident into node *j*
  - H_{ij}= 0 otherwise

Node 1 ~ 4

Edge e_1 ~ e_5

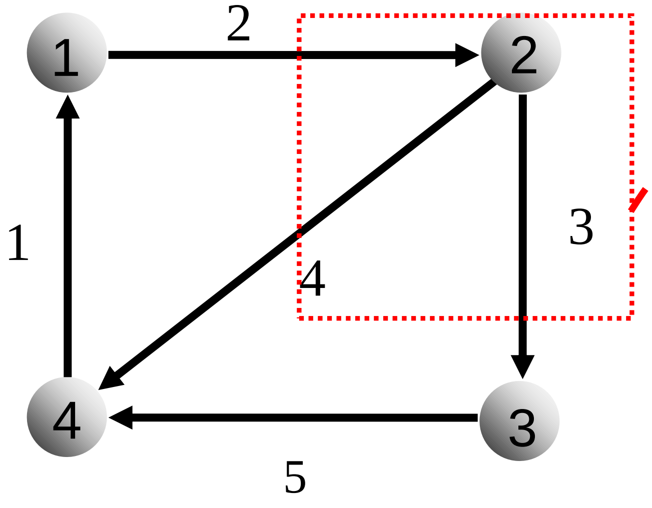
	1	2	3	4
e_1	1	-1	0	0
e_2	-1	0	0	1
e_3	0	1	0	-1
e_4	0	1	-1	0
e_5	0	0	1	-1

Laplacian matrix for *directed* graph

- *Laplacian matrix* for a directed graph: L

$$L = D - A$$

  - D_{ii} : sum of the weights (out-degrees) of all the out-going edges from i
  - A : adjacency matrix



Directed graph G

Node 1 ~ 4

Node 1 ~ 4

	1	2	3	4
1	**2**	-2	0	0
2	0	**7**	-3	-4
3	0	0	**5**	-5
4	-1	0	0	**1**

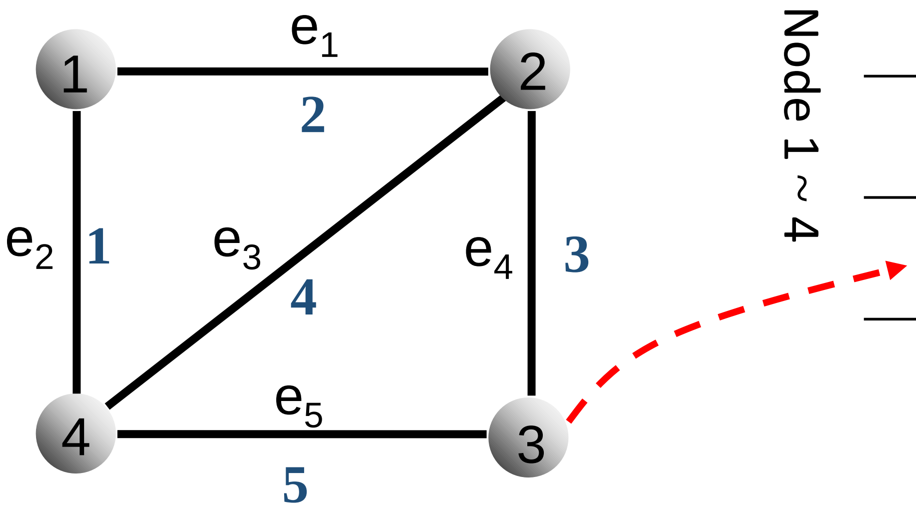
L is asymmetric,
and has zero row sums

Laplacian matrix for *undirected* graph

- *Laplacian matrix* for an undirected graph:

$$L = D - A = H^{\mathrm{T}} \mathbf{W} H$$

  - D_{ii} : sum of the weights of all the edges associated with i
  - A : adjacency matrix
  - H : incidence matrix
  - $\mathbf{W}$: diagonal weight matrix
  - $\mathbf{W}_{ii}$: (positive) edge weight of e_i



Undirected graph *G*

Node 1 ~ 4

Node 1 ~ 4

	1	2	3	4
1	**3**	-2	0	-1
2	-2	**9**	-3	-4
3	0	-3	**8**	-5
4	-1	-4	-5	**10**

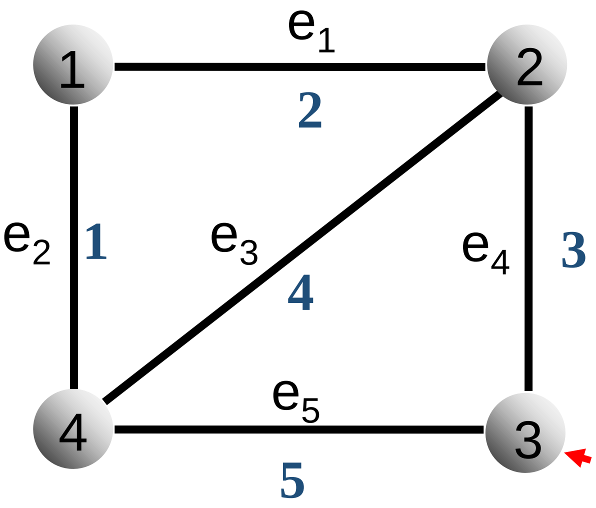
L is symmetric,
has zero row sums
and zero column sums

Laplacian matrix for *undirected* graph

- *Laplacian matrix* of an undirected graph:

$$L = D - A = H^{T}\, \mathbf{W}\, H$$

  - *The two definitions are equivalent.*
  - *The second definition $L = H^{T} W H$ implies that L does not depend on the labeling and orientation chosen for the graph in defining H.*



Undirected graph *G*

Node 1 ~ 4

	1	2	3	4
1	**3**	-2	0	-1
2	-2	**9**	-3	-4
3	0	-3	**8**	-5
4	-1	-4	-5	**10**

Since L is symmetric, all its eigenvalues are real

Linear algebra and graph Laplacian (undirected graphs)

For Laplacian L, we can order the (real) eigenvalues in an ascending order

$$0 \leq \lambda_1(L) \leq \lambda_2(L) \leq . . . \leq \lambda_n(L)$$

In this case, λ_k refers to the k-th smallest eigenvalue of the (undirected graph) Laplacian

- By construction, $L(\mathrm{G})\mathbf{1} = 0$ for any graph ----> So $\lambda_1(L) = 0$.
- When is $\lambda_2(L) > 0$?
- In other words, we need to characterize the null space of $L(\mathrm{G})$:

$$\mathrm{Null}\,(L(\mathrm{G})) = \{z \in \mathbf{R}^n \mid L(\mathrm{G})z = 0\}$$

What are the vectors in Null (L(G)) except the subspace generated by $\underline{\mathbf{1} = [1, 1, ..., 1]^T}$?

Linear algebra and undirected graph Laplacian: null spaces

In order to answer this question, notice that if $z \in \mathrm{Null}\,(L(G))$, then

$$L(G)z = H^T(G)\,H(G)\,z = 0$$

which is equivalent to

$$H(G)\,z = 0 \;(\text{ Note: } \mathrm{Null}(L) = \mathrm{Null}(H)\;)$$

This means that if $(i, j) \in E$, then $z_i = z_j$; so, if the graph is connected,

$$z_1 = z_2 = \ldots = z_n$$

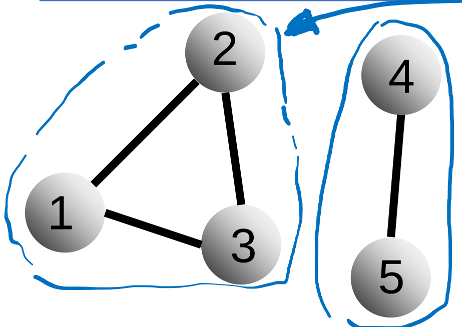
That is $z = \alpha\mathbf{1}$ for some α!

$$H = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

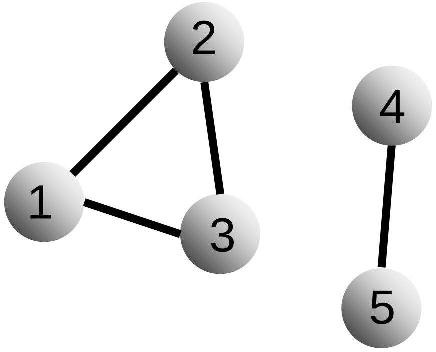
$$z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{bmatrix}, \quad Hz = \begin{bmatrix} z_1 - z_2 \\ z_2 - z_3 \\ z_1 - z_3 \\ z_4 - z_5 \end{bmatrix} = 0$$

$$z_1 = z_2 = z_3$$

$$z_4 = z_5$$



Linear algebra and undirected graph Laplacian: null spaces



In order to answer this question, notice that if $z \in \mathrm{Null}\,(L(\mathrm{G}))$, then

$$L(\mathrm{G})z = H^{T}(\mathrm{G})\ H(\mathrm{G})\,z = 0$$

which is equivalent to

$$H(\mathrm{G})\,z = 0 \ (\ \text{Note: } \mathrm{Null}(L) = \mathrm{Null}(H)\ )$$

This means that if $(i, j) \in E$, then $z_i = z_j$; so, if the graph is connected,

$$H = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

$$z_1 = z_2 = \ldots = z_n$$

That is $z = \alpha\mathbf{1}$ for some α!

Actually, z_i is constant on each (connected) component of G.

- Every connected component introduces an additional dimension in $\mathrm{Null}(L(\mathrm{G}))$.

Theorem

Let G have c connected components (when $c = 1$ the graph is connected).
Then rank $L(G) = n - c$.

The above theorem also holds for *weighted* (undirected) graphs.

Graph Laplacian (undirected graphs): Rank, λ_2, and connectivity

Rank condition:

rank $L(G) = n - 1$ if and only if G is connected.

Equivalently (eigenvalue condition)

G is connected if and only if $\lambda_2(L) > 0$.

The eigenvalue λ_2 captures a quantitative notion of graph connectivity: more positive λ_2 means the graph is more connected

- Adding more edges increases λ_2
- Increasing edge weights also increases λ_2

Courant-Fisher formulation (Note the Laplacian L is positive semi-definite)

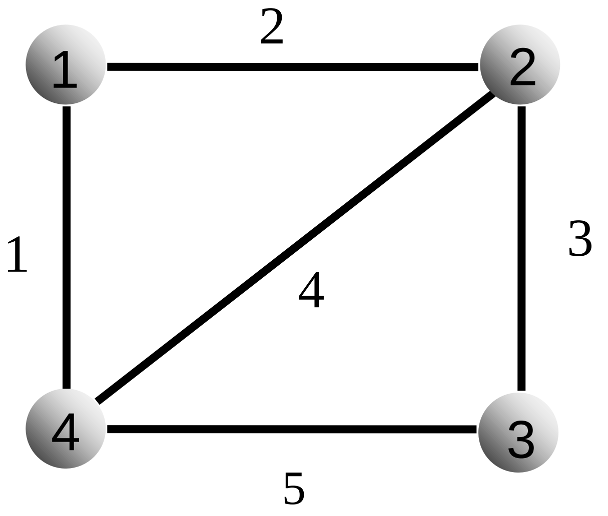
$$\lambda_2(G) = \min_{x \perp \mathbf{1}, ||x|| = 1} x^T L(G)\, x$$

Equivalently

$$\lambda_2(G) \leq x^T L(G)\, x, \text{ for all } x \perp \mathbf{1},\ ||x|| = 1$$

Properties of Laplacian matrix: **undirected graph**

A *connected undirected* graph



Undirected graph *G*

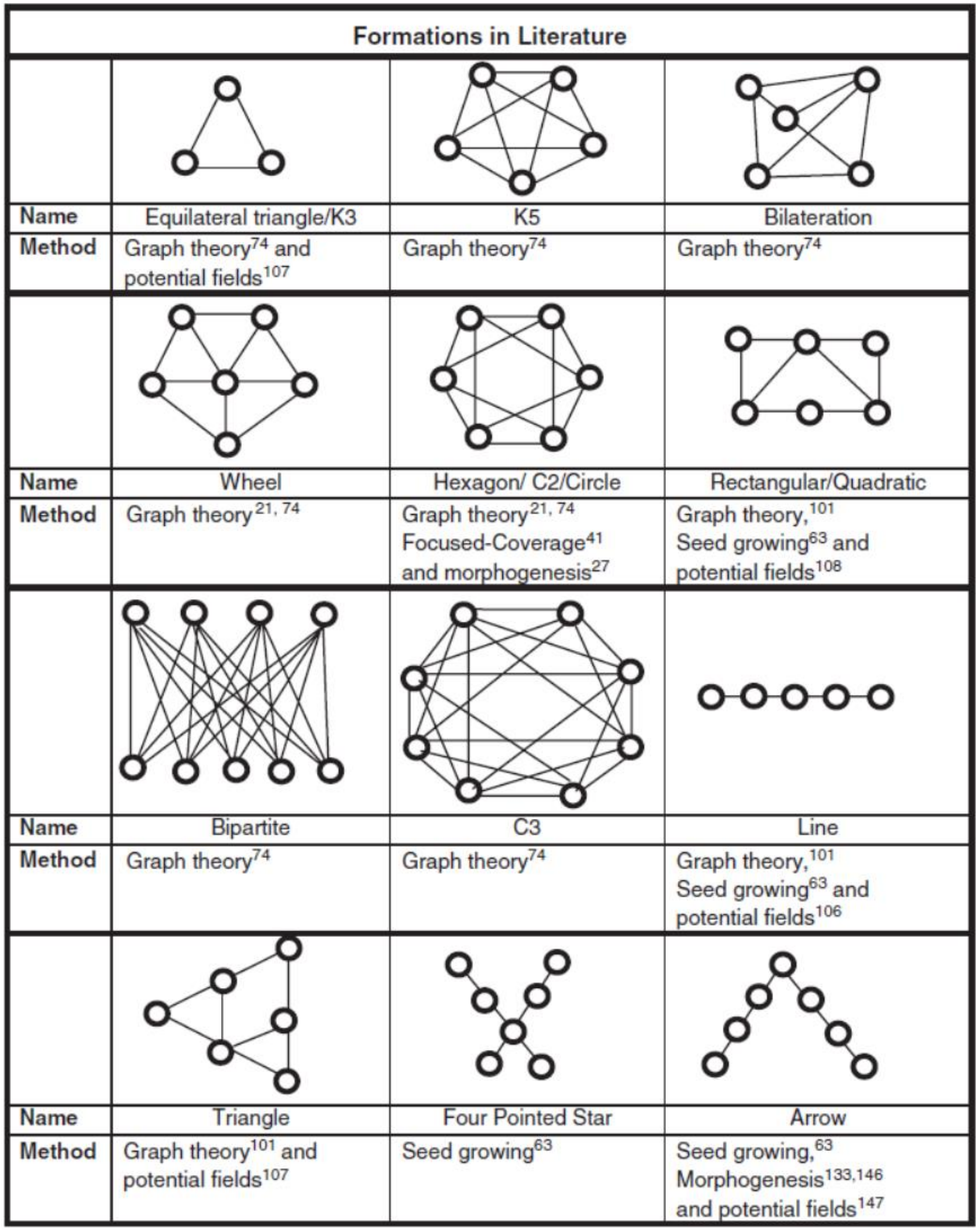
$$L = \begin{bmatrix} 3 & -2 & 0 & -1 \\ -2 & 9 & -3 & -4 \\ 0 & -3 & 8 & -5 \\ -1 & -4 & -5 & 10 \end{bmatrix}$$

- $L = L^\top$, i.e., it is symmetric
- $L\mathbb{1} = 0, \mathbb{1}^\top L = 0,$ i.e., both the row and column sums are zero
- L is positive semi-definite
- For connected graphs, rank $L = n - 1$; L has only one zero eigenvalue, and all the other eigenvalues are positive

$$\mathrm{eig}(L): \quad 0 \quad \boxed{3.6595} \quad 11.8104 \quad 14.5300$$

$\uparrow$

λ_2



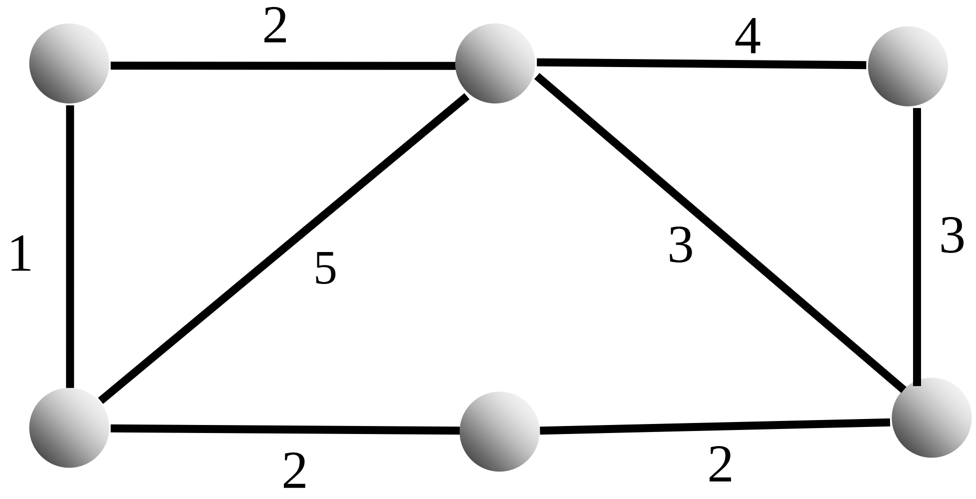
Formations in Literature

Name	Equilateral triangle/K3	K5	Bilateration
Method	Graph theory[74] and potential fields[107]	Graph theory[74]	Graph theory[74]
Name	Wheel	Hexagon/ C2/Circle	Rectangular/Quadratic
Method	Graph theory[21, 74]	Graph theory[21, 74] Focused-Coverage[41] and morphogenesis[27]	Graph theory,[101] Seed growing[63] and potential fields[108]
Name	Bipartite	C3	Line
Method	Graph theory[74]	Graph theory[74]	Graph theory,[101] Seed growing[63] and potential fields[106]
Name	Triangle	Four Pointed Star	Arrow
Method	Graph theory[101] and potential fields[107]	Seed growing[63]	Seed growing,[63] Morphogenesis[133,146] and potential fields[147]

- Barca, J.C. and Sekercioglu, Y.A., 2013. Swarm robotics reviewed. *Robotica*, *31*(3), pp.345-359.

Online quiz

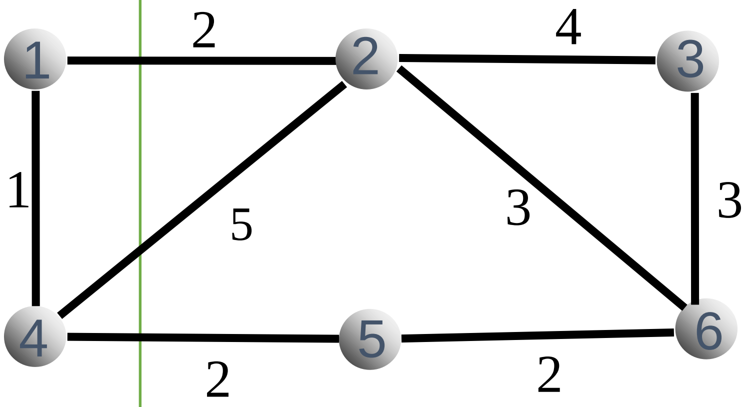
What is the value of the second smallest eigenvalue (λ_2) of the Laplacian matrix for the following weighted undirected graph?

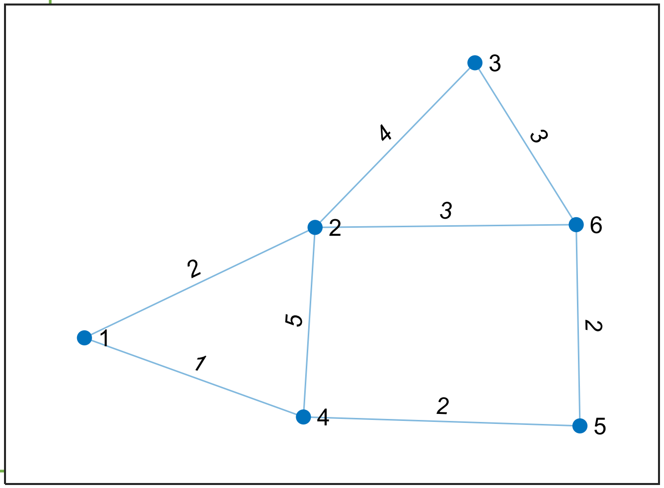


(a) 1.3820
(b) 2.9182
(c) 1.9186
(d) 2.2256

Graph in Matlab

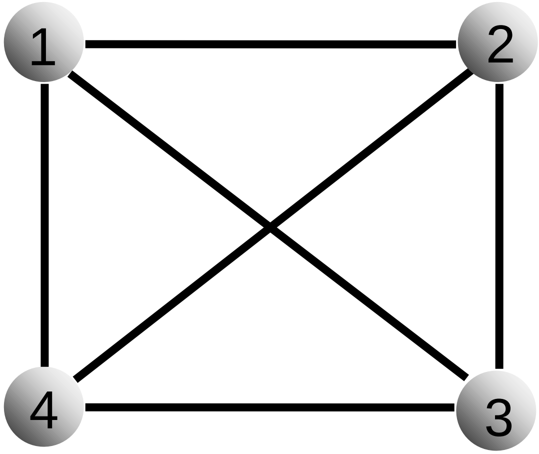
```
%% define a graph
% starting nodes
s = [1 1 2 2 2 3 4 5 ];
% terminal nodes
t = [2 4 3 4 6 6 5 6 ];
% edge weights
weights = [2 1 4 5 3 3 2 2];

%% plot a graph
% label/name each node
names = {'1' '2' '3' '4' '5' '6'};
G = graph(s,t,weights,names);
plot(G,'EdgeLabel',G.Edges.Weight) % plot the graph


%% matrix representations
% Incidence matrix (convention: row-nodes and column-edges)
I = incidence(G);
% Laplacian (unweighted)
L = laplacian(G);


W = diag([2 1 4 5 3 3 2 2]);
% Construct weighted Laplacian
L_w = I*W*I';
% eigenvalues
eig(L_w)
```





Spectra of Laplacian matrix: special **undirected graphs**

- Complete graph (K_4 graph)



A complete undirected graph *G*

$$L = \begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix}$$

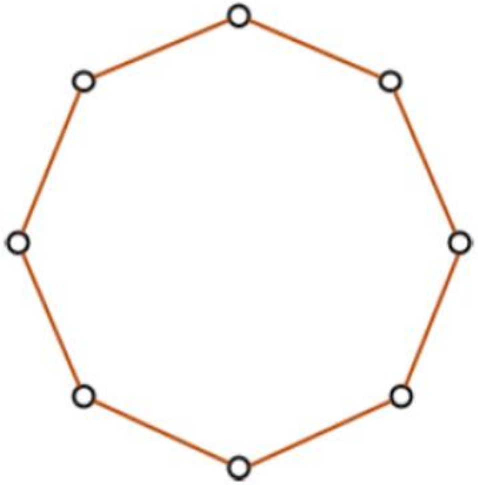
$$\boxed{\text{eig}(L): \quad 0 \quad 4 \quad 4 \quad 4}$$

- A general complete graph (K_n graph with *n* nodes and *n(n+1)/2* edges)

$$L(K_n) = \begin{bmatrix} n-1 & -1 & \cdots & -1 & -1 \\ -1 & n-1 & \cdots & -1 & -1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -1 & -1 & -1 & -1 & n-1 \end{bmatrix} = nI - \mathbf{1}\mathbf{1}^T$$

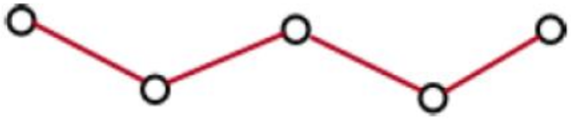
$$\boxed{\text{eig}(L): \quad 0, \quad n, \quad n, \quad \ldots\ldots \quad n}$$

Spectra of Laplacian matrix: **special undirected graphs**



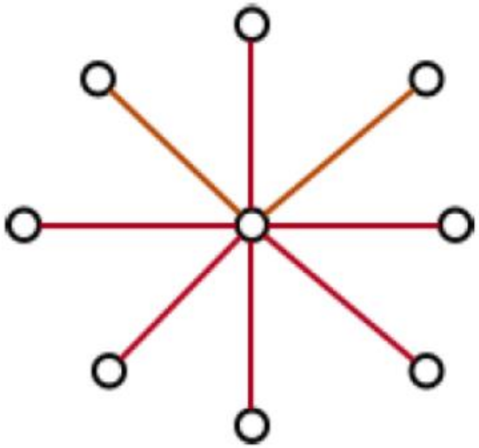
Cycle graph: *n* nodes and *n* edges

eig(L): $2\left(1-\cos 2k\pi/n\right), \quad k=0,1,\ldots n-1$



Line graph: *n* nodes and *(n-1)* edges

eig(L): $2\left(1-\cos k\pi/n\right), \quad k=0,1,\ldots n-1$

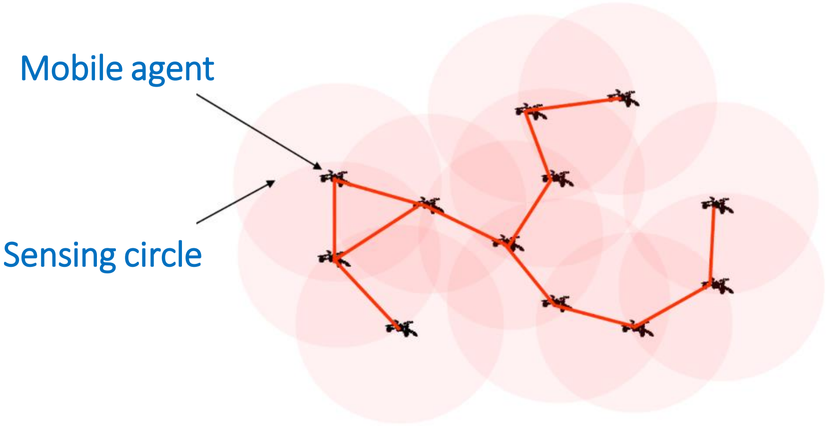


Star graph: *n* nodes and *(n-1)* edges

eig(L): $n-2$ eigenvalues of 1, one eigenvalue of zero (as always) and last one is $2(n-1)-(n-2)=n$

Our first distributed algorithm: Consensus

Multi-agent consensus & flocking/schooling



Fish schooling

Multi-agent consensus protocol

Formulation of the consensus problem

- Consider *N* agents with an internal state $x_i \in Rd$
  - For now, we consider the scalar state case (*d=1*). Multi-dimensional state case can be modelled via Kronecker products.

- Consider an internal dynamics for the state evolution
  - For now, we consider single integrator dynamics $\dot{x}_i = ui$

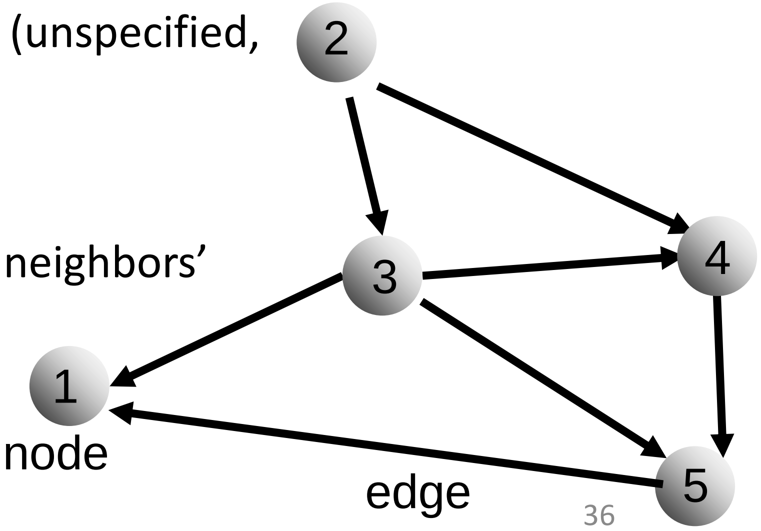
- Consider an interaction graph G,
  - Having the agents as nodes, and sensing/communication links as edges

Problem: design the control inputs u_i so that

- All the states x_i agree on the same common value $\bar{x}$ (unspecified, e.g., average value)

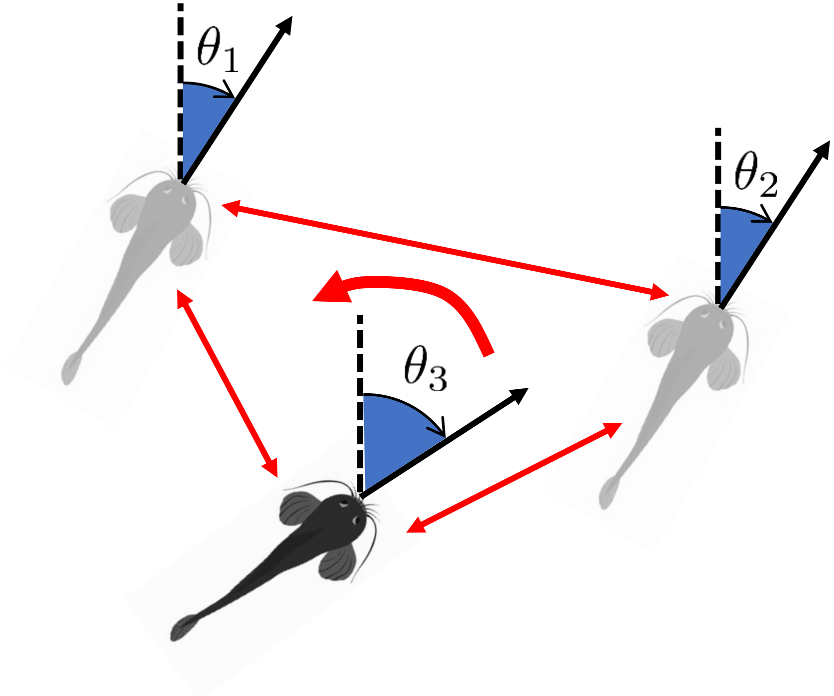
$$\lim_{t \to \infty} x_i(t) = \bar{x}, \ \forall i$$

- By making use of only relative information w.r.t. the neighbors' state (relative sensing and decentralization)



Example: Multi-agent consensus

- To model the fish schooling behavior, each fish steers towards the average heading of its neighbors.



The center fish rotates anti-clockwise to align itself with the average heading of its neighbors

$$\dot{\theta}_3 = \tfrac{1}{2}(\theta_1 - \theta_3) + \tfrac{1}{2}(\theta_2 - \theta_3)$$

Applying this rule to the other two:

$$\dot{\theta}_1 = \tfrac{1}{2}(\theta_2 - \theta_1) + \tfrac{1}{2}(\theta_3 - \theta_1)$$

$$\dot{\theta}_2 = \tfrac{1}{2}(\theta_1 - \theta_2) + \tfrac{1}{2}(\theta_3 - \theta_2)$$

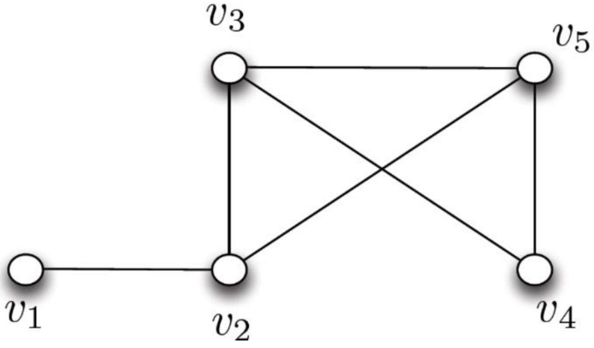
$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = -\tfrac{1}{2} \underbrace{\boxed{\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}}}_{L} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

Laplacian matrix of an undirected network

Example: Multi-agent consensus

- Consider a multi-agent system in the graph (node state x_i)
- Consensus controller: let u_i be the sum of all the differences of the neighbors' states w.r.t. the state of agent i

$$\begin{aligned}
u_1 &= (x_2 - x_1)\\
u_2 &= (x_1 - x_2) + (x_3 - x_2) + (x_5 - x_2)\\
u_3 &= (x_2 - x_3) + (x_4 - x_3) + (x_5 - x_3)\\
u_4 &= (x_3 - x_4) + (x_5 - x_4)\\
u_5 &= (x_2 - x_5) + (x_3 - x_5) + (x_4 - x_5)
\end{aligned}$$



- In compact form for all the agents

$$u = -\underbrace{\boxed{\begin{bmatrix} 1 & -1 & 0 & 0 & 0\\ -1 & 3 & -1 & 0 & -1\\ 0 & -1 & 3 & -1 & -1\\ 0 & 0 & -1 & 2 & -1\\ 0 & -1 & -1 & -1 & 3 \end{bmatrix}}}_{L} x$$

Laplacian matrix of an undirected network

$$\boxed{\dot{x} = -Lx}$$

Consensus dynamics

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = -\frac{1}{2} \underbrace{\boxed{\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}}}_{L} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} = -\underbrace{\boxed{\begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 \\ 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & -1 & 3 \end{bmatrix}}}_{L} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$$

Consensus dynamics

$$\dot{x}(t) = -L(G)x(t)$$

$$\boxed{\dot{x}_i(t) = -\sum_{j \in N_i} w_{ij}(x_i(t) - x_j(t))}$$

L: the (weighted) Laplacian matrix of an undirected network

w_{ij}: the weight in the edge (i, j)

Convergence of consensus dynamics (undirected graph)

Consensus dynamics

$$\dot{x}(t) = -L(G)x(t)$$

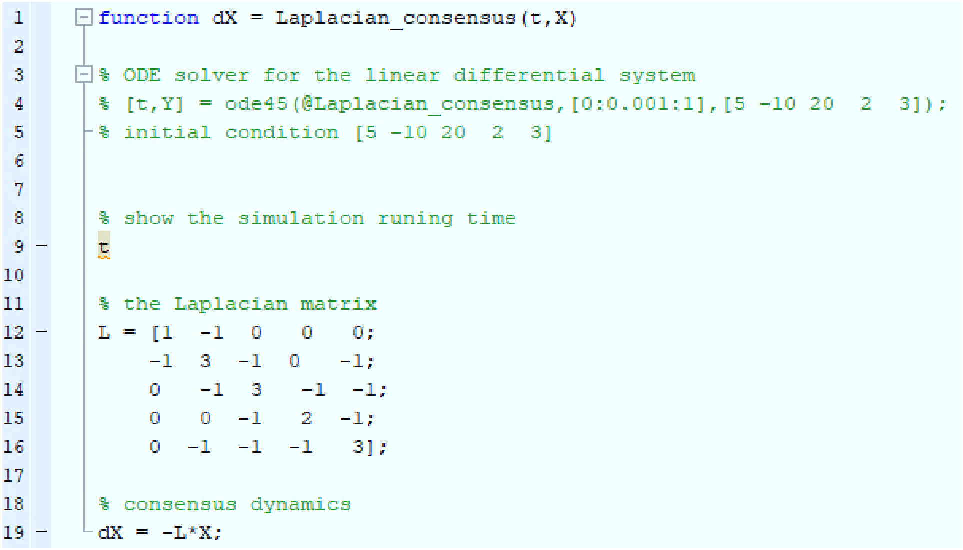
Theorem

Consider the consensus dynamics under an undirected and connected graph with the (weighted) Laplacian *L*.
(1) All node states reach an average consensus.
(2) The convergence rate is characterized by $\boldsymbol{\lambda_2}$ (the algebraic connectivity).

$$\boxed{x(t) \to \frac{\mathbf{1}^T x(0)}{n}\mathbf{1}}$$ at a rate proportional to $\boldsymbol{\lambda_2}$!

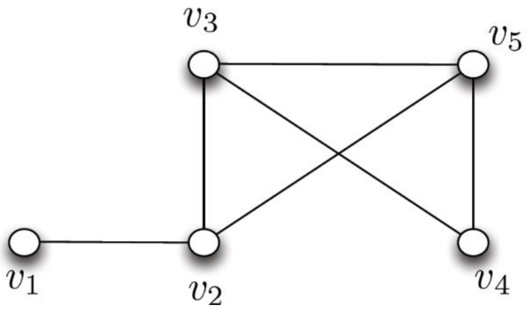
$$1^T x_0 = [1,1,\cdots,1]\begin{bmatrix} x_1(0) \\ x_2(0) \\ \vdots \\ x_N(0) \end{bmatrix} = x_1(0)+x_2(0)+\cdots+x_N(0)$$

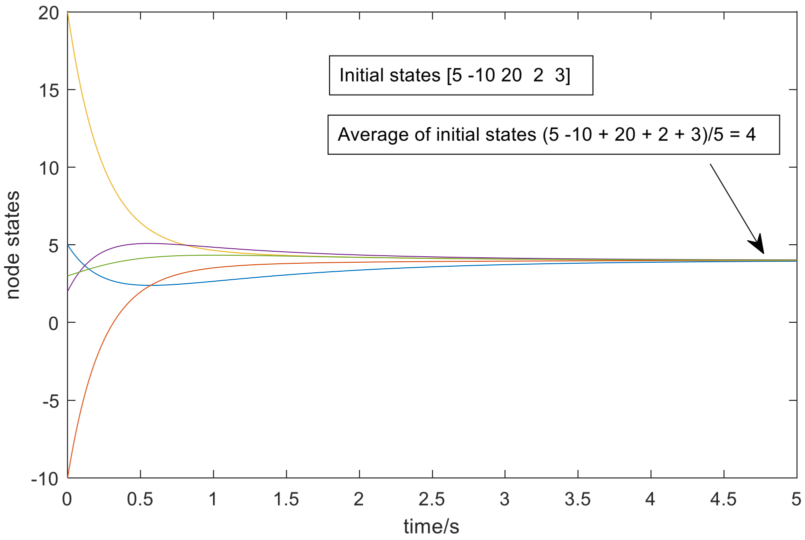
Convergence of consensus dynamics (undirected graph)



```
function dX = Laplacian_consensus(t,X)

% ODE solver for the linear differential system
% [t,Y] = ode45(@Laplacian_consensus,[0:0.001:1],[5 -10 20  2  3]);
% initial condition [5 -10 20  2  3]


% show the simulation runing time
t

% the Laplacian matrix
L = [1  -1  0    0    0;
    -1  3  -1  0   -1;
    0   -1  3   -1  -1;
    0   0  -1   2  -1;
    0  -1  -1  -1   3];

% consensus dynamics
dX = -L*X;
```

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} = -\begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 \\ 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & -1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$$





Initial states [5 -10 20 2 3]

Average of initial states (5 -10 + 20 + 2 + 3)/5 = 4

Summary/take home messages

- Complex network control system by graph modelling
- Basic graph theory: undirected/directed/weighted graphs
- Graph and matrices: adjacency/incidence/Laplacian matrix
- Graph Laplacian and algebraic connectivity ($\lambda_2(L)$)
- Introduction to multi-agent consensus dynamics (undirected graphs)

The next lecture

- Multi-agent consensus: theory, proofs, and applications