Structure Learning in Infrastructure Networks

Rajasekhar Anguluri

Department of Computer Science and Electrical Engineering

University of Maryland, Baltimore County (UMBC)

rajangul@umbc.edu

Summer Workshop, July 15 – 18, 2024

IIT, Bombay

slides template inspiration: Joel Tropp, Caltech

https://rajanguluri.github.io

My Research Interests

- **inverse problems**: parameter or signal estimation problems in networks and control systems

- **matrices in action**: matrices arising in applications (e.g. ML, data science, and controls)

- **energy systems**: formulating (and solving) problems that have linear algebraic or systems theory flavor

I show off to appear serious…but I am not…so talk to me!

Learning vs Estimation vs Identification

- **estimating**: estimate parameters of a model structure using data
- **learning**: refine models (tuning parameters) that can generalize from data
- **identifying**: find models (and parameters) to find the system generating data

In a Platonic sense, and for this workshop, all three are the same

source: my recent friend ChatGPT

Technical Jargon (Synonyms)

- **nodes:** buses (power system terminology)
- **structure:** topology or Laplacian matrix
- **learning:** estimation or identification
- **latent:** hidden or un-observed
- **complex-network:** network with complex-valued edge weights
- **currents:** flows;
- **voltages:** potentials
- **conservation:** equilibrium or balance

Welcome!

- **Motto:** introduce to estimation problems at the intersection of networks, optimization, and statistical learning
  - **Day 1:** modeling infrastructure networks
  - **Day 2:** structure estimation in infrastructure networks
  - **Day 3:** sparse modeling: theory and algorithms
  - **Day 4:** structure estimation via sparse optimization

What are you in for?

for:

- modeling
- interpreting
- implementing

not for:

- rigorous derivations and proofs
- solving “real” engineering problems
- programming

After this course…you should

- use Linear Algebraic tools to model infrastructure networks
- use modern statistical tools for estimation problems
- use alternating direction method of multipliers
- be excited to learn more about “not abouts”

Day 1: Overview

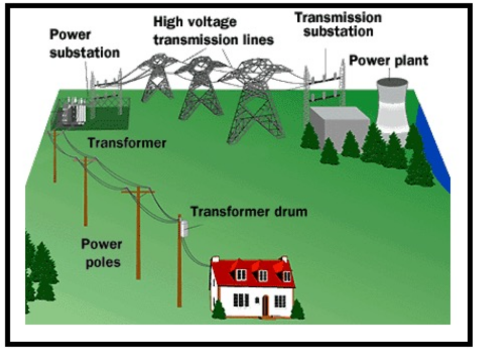
- critical infrastructure networks (CIN)
- equilibrium equations for CIN
- structure learning problems in CIN
- structure learning via linear models
- wrap up

Critical Infrastructure Networks (CIN)

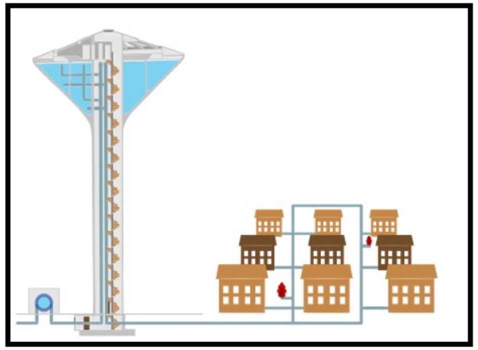
- CIN keep our society running
- Dept. of homeland security (USA) identified 16 critical sectors:

  chemical, food, energy, financial, defense, healthcare, communications, etc.

- **energy systems:** uses energy to meet our needs



power system



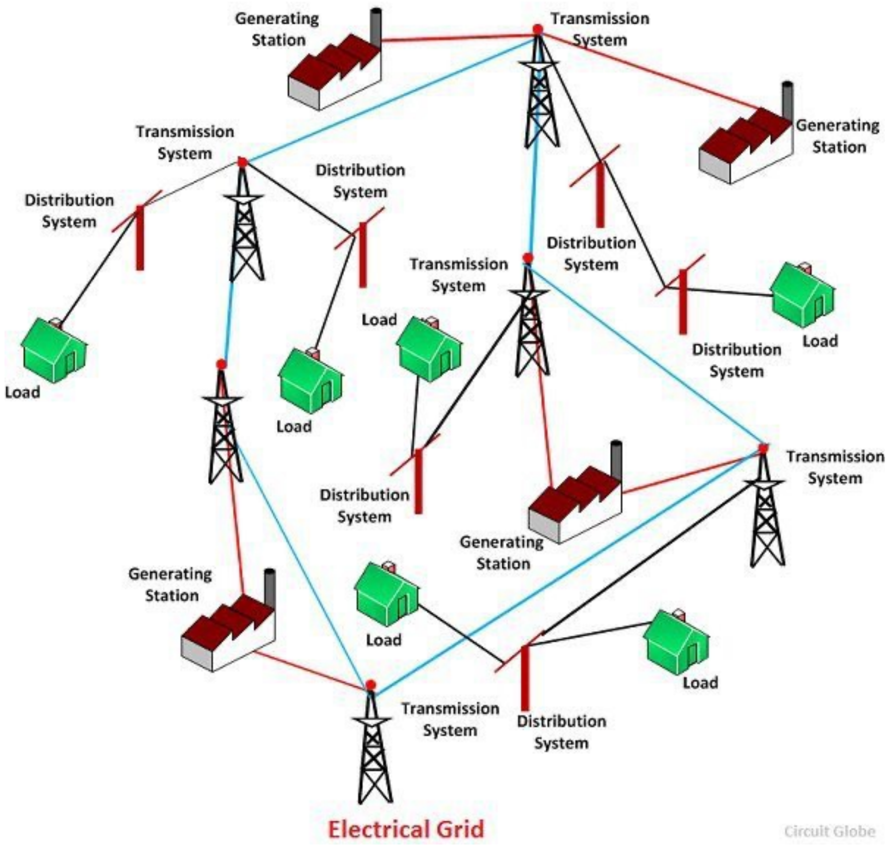
water supply system

gas/oil system

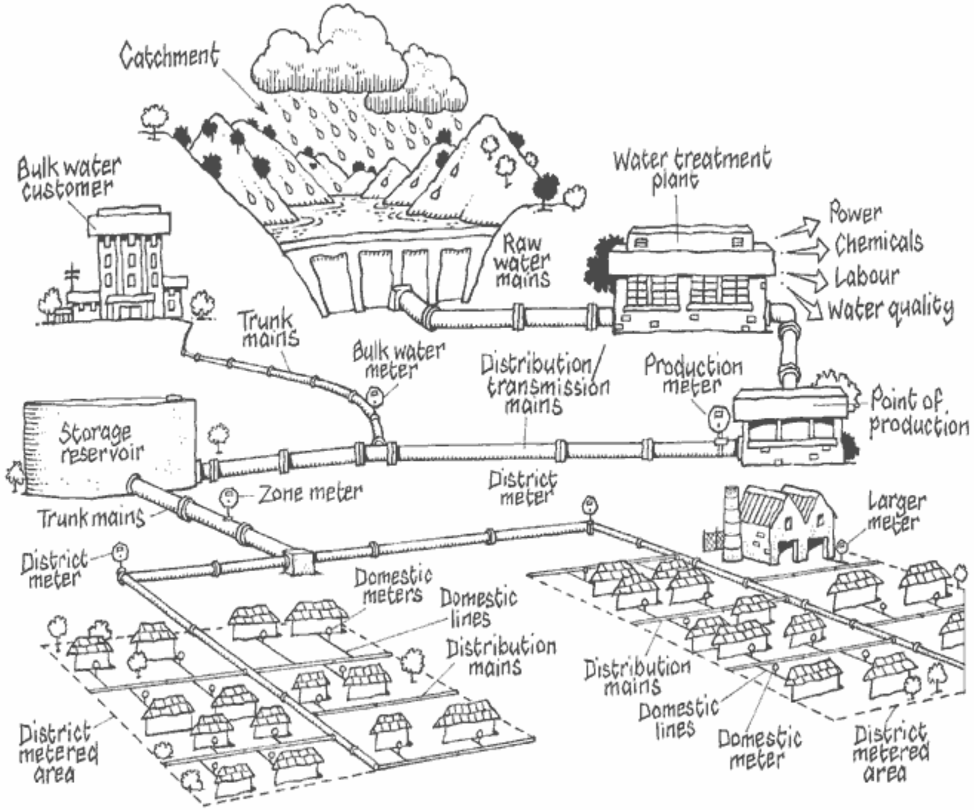
figure source: web

Critical Infrastructure Networks (CIN)

critical infrastructure systems are realized through "network" of interconnected systems



network structure in power systems

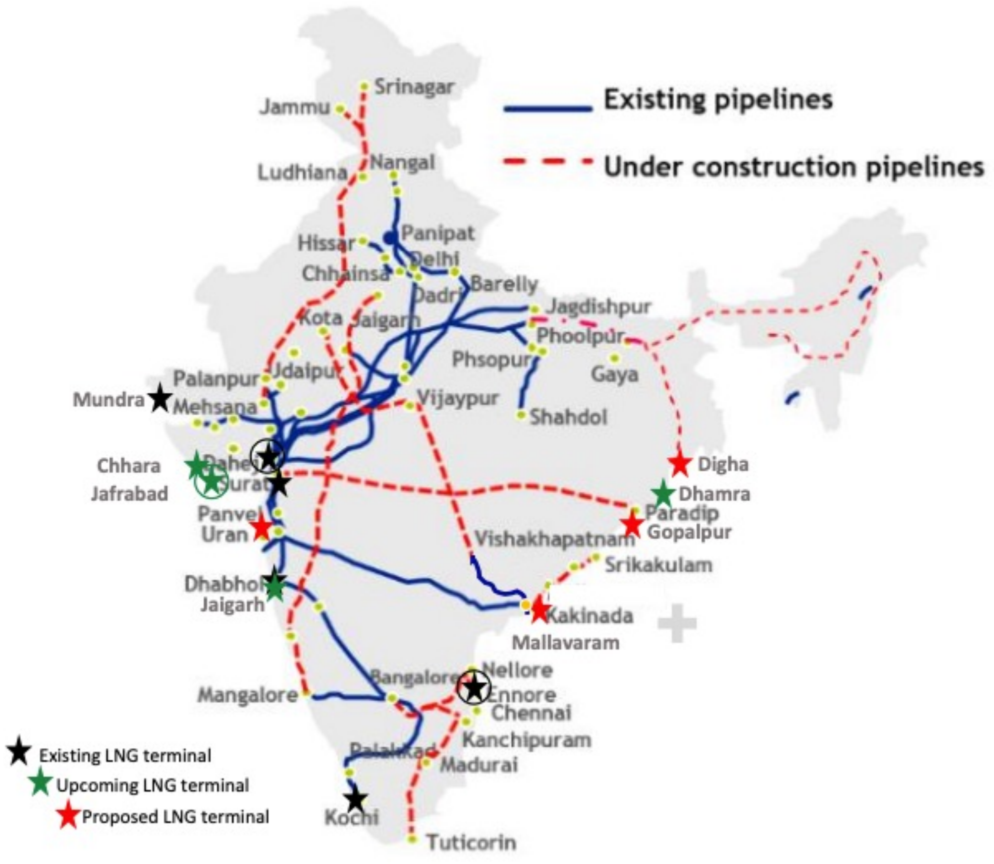


network structure in water distribution system

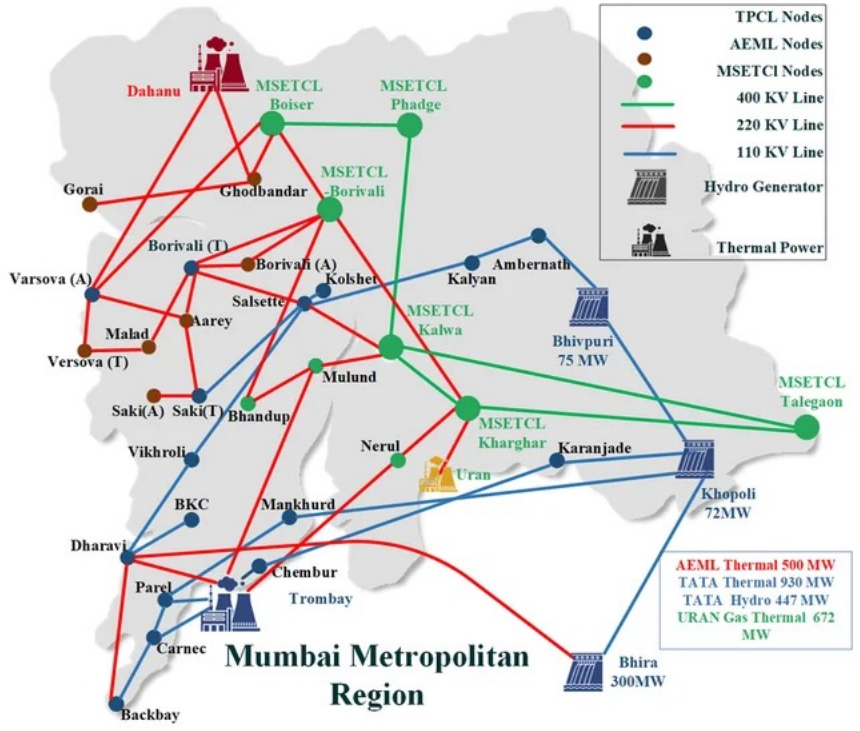
figure source: web

Examples of “realistic” Networks

critical infrastructure systems are realized through ”network” of interconnected systems



natural gas line pipe network

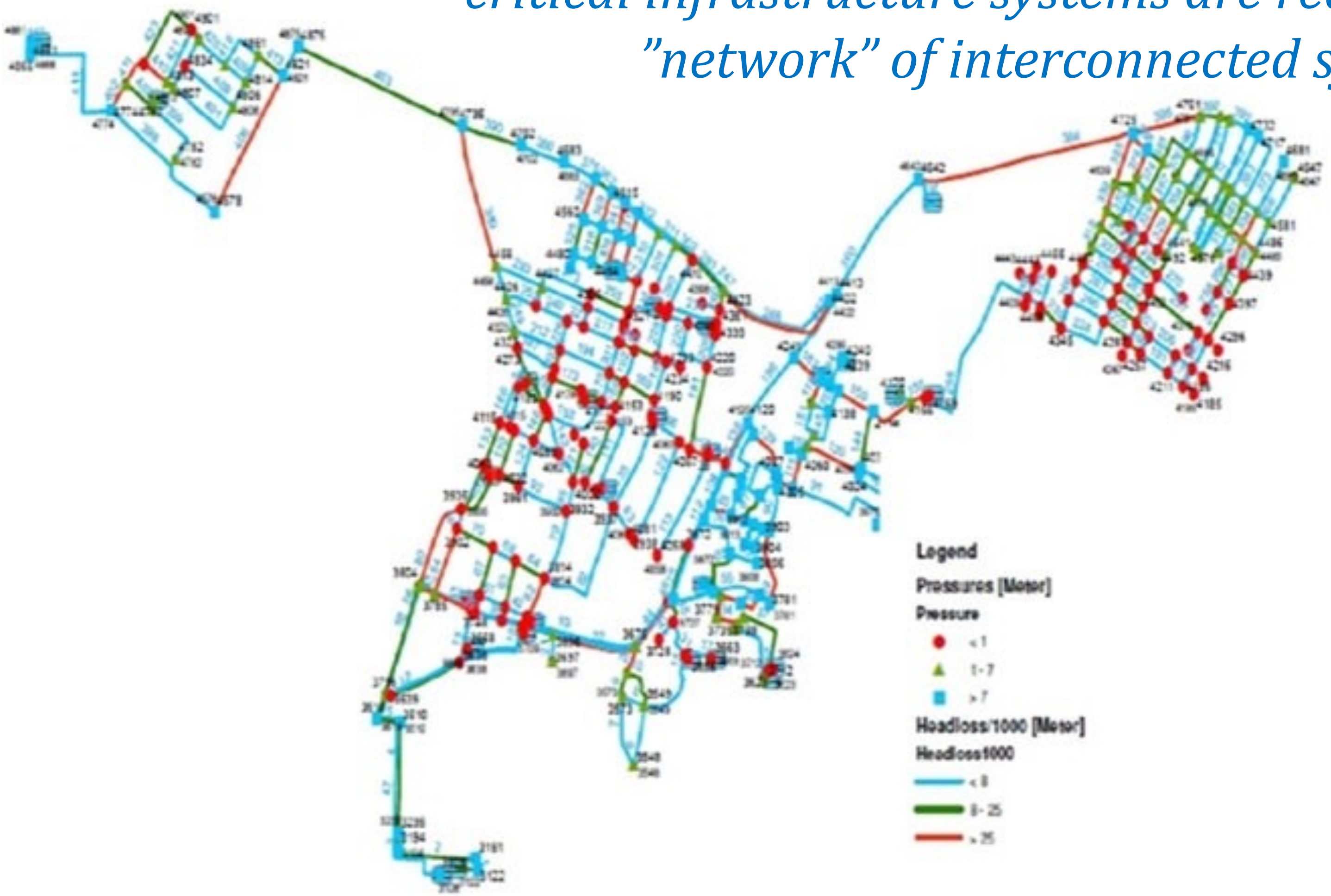


high-voltage transmission network

figure source: web

Examples of “realistic” Networks

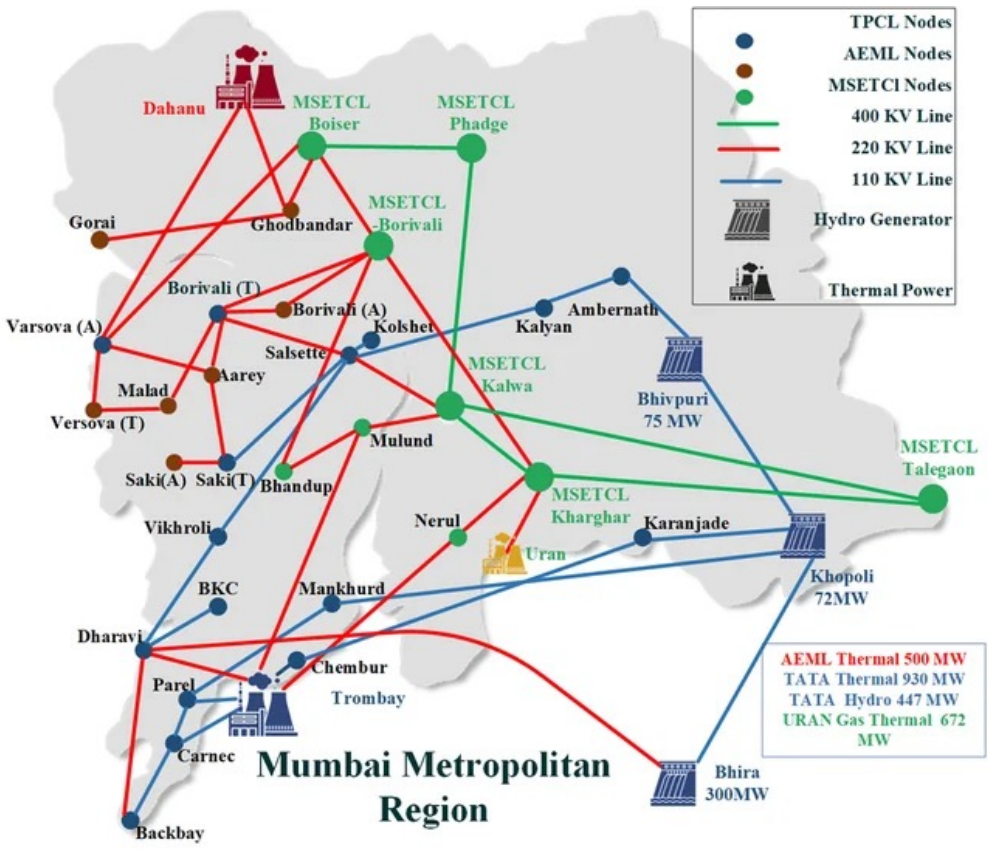
critical infrastructure systems are realized through ”network” of interconnected systems



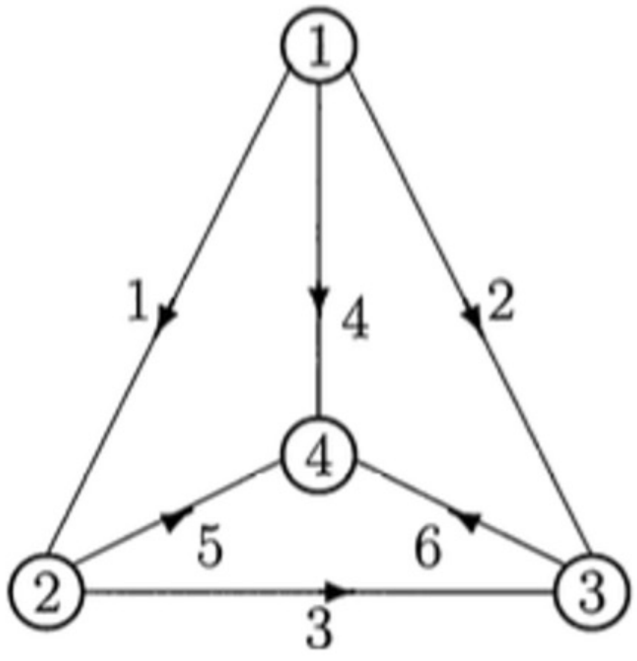
Rajasthan’s water distribution network

figure source: web

Abstract Representation of CINs



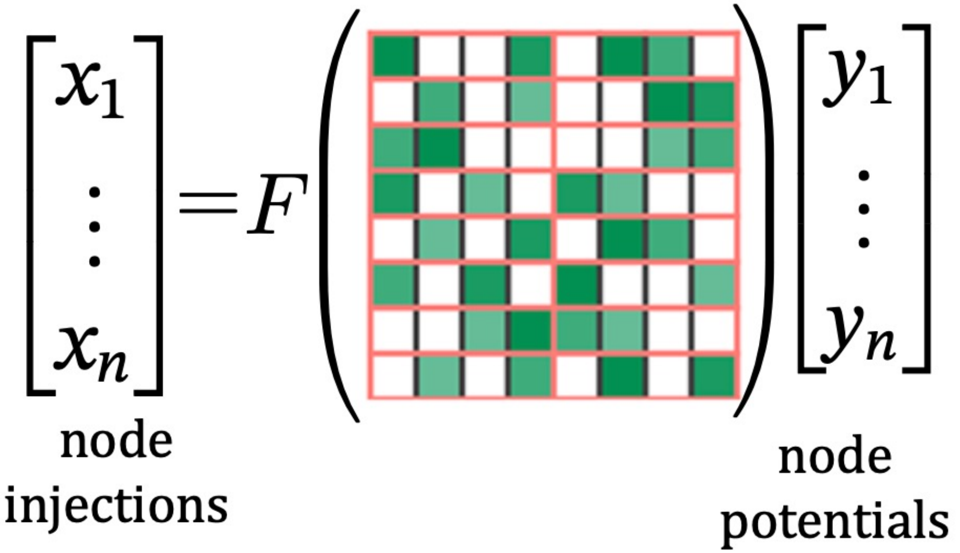
network structure in power systems



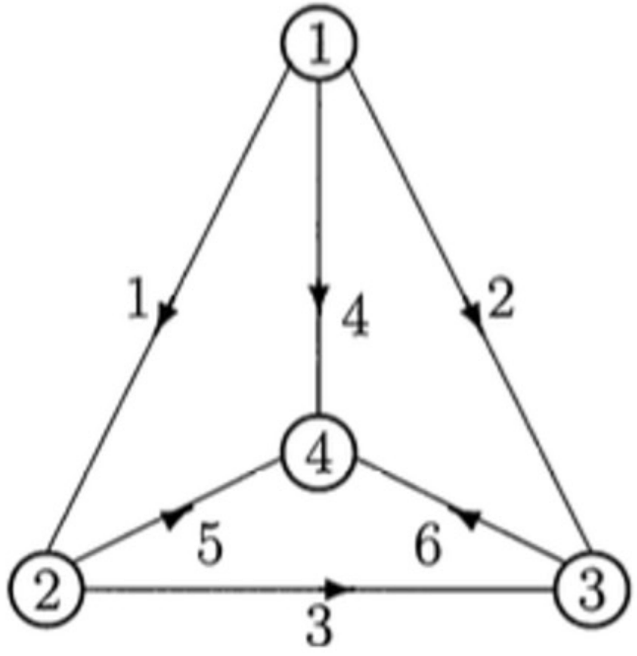
abstract network representation

Equilibrium Equations for CIN

conservation law: flow in = flow out



(i) equilibrium equation



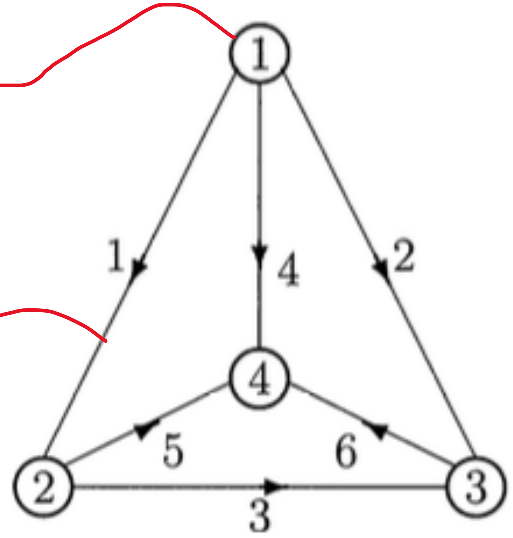
(ii) Infrastructure network

❧ direction of edges in this representation is arbitrary (arrow is always pointed to node with a higher label)

Incidence Matrix and Potentials

- **graph:** contains set of nodes (p) and set of edges (m) between them
- **incidence matrix:** matrix $A \in \{-1,0,1\}^{m \times p}$; gives connectivity information

node



edge

node

$$A = \begin{array}{c} \begin{array}{cccc} ① & ② & ③ & ④ \end{array} \\ \left[\begin{array}{rrrr} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \end{array}\right] \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array}$$

edges 1 to 6

Incidence matrix

complete graph with $m = 6$ edges and $p = 4$ nodes

Incidence Matrix and Potentials

- the incidence matrix acts as a *difference matrix:*

(differences across edges)

$$Ay = \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} y_2 - y_1 \\ y_3 - y_1 \\ y_3 - y_2 \\ y_4 - y_1 \\ y_4 - y_2 \\ y_4 - y_3 \end{bmatrix}$$

- the row vector $y = (y_1, \dots, y_p)$ is called the **node potential**

leveling (geodesy)	heights	height difference
water	pressures	pressure difference
power	voltages	voltage difference
general	potentials	**potential difference**

$$y = \begin{cases} \text{heights} \\ \text{pressures} \\ \text{voltages} \\ \text{potentials} \end{cases} \qquad Ay = \begin{cases} \text{height difference} \\ \text{pressure difference} \\ \text{voltage difference} \\ \textbf{potential difference} \end{cases}$$

Incidence Matrix and Potentials

- **null space of A**: what combination of the columns of A are such that $Ay = 0$?

$$A = \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \quad \text{with} \quad y = c\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \quad \text{will satisfy} \quad Ay = 0$$

- Rank(A)=n−1 implies $Ay = b$ cannot be solved uniquely (intuition: we can raise all the voltages by a constant unit, and the edge differences will be the same)

- dimension of Null(A)=1 implies graph is connected (exercise: if disconnected what is Null(A)?)

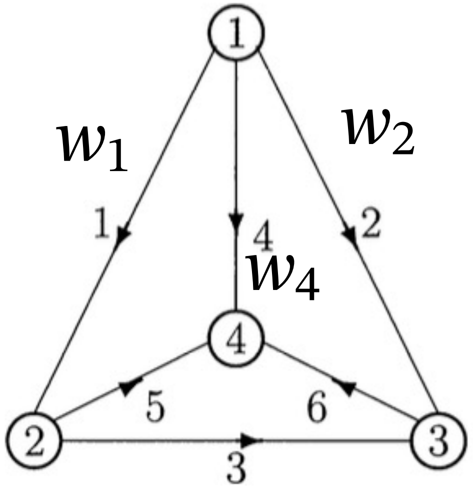
- reduced-order A: (i) add a self-loop; (ii) remove a column (set $y_p = 0$)

Kirchoff's Current Law (KCL)

- **KCL**: conservation law for **edge currents** $w = [w_1, \dots, w_m]^T$
- *Matrix notation*: *sum of edge currents at a node is zero* $(A^T w = 0)$

$$A^{\mathrm{T}} = \begin{bmatrix} -1 & -1 & 0 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

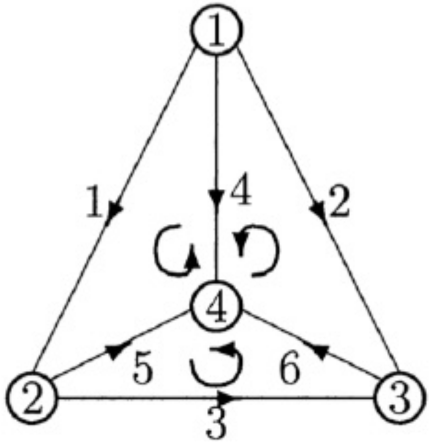
$-w_1 - w_2 - w_4 = 0$	node 1
$w_1 - w_3 - w_5 = 0$	node 2
$w_2 + w_3 - w_6 = 0$	node 3
$w_4 + w_5 + w_6 = 0$	node 4



- KCL is also referred to as *law of continuity*
- $A^T w = x$: currents (flows) that "balance themselves" <u>*with*</u> external **nodal injection**
- $1^T (A^T) = 0$ implies $1^T x$=0 (that is, sum of external flows sum to zero)

Kirchoff's Voltage Law (KVL)

- **KVL**: sum of potential differences over a loop is zero
- *Matrix language* : let $e = [e_1, \dots, e_m]^T$ be the **potential difference**, then $e = Ay$



Loop flows $w = \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \\ 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ -1 \\ 1 \end{bmatrix}$

loop currents (flows) solve KCL: $A^T w = 0$

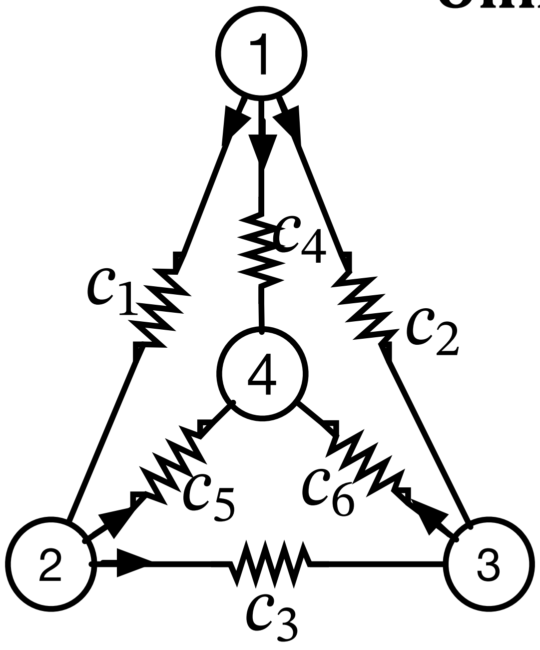
- sum of differences: $e^T w$
- KVL states: $e^T w = 0$
- Fundamental theorem of LA:

$$\mathrm{Range}(A) \perp \mathrm{Null}(A^T)$$

Main point: if $A^T w = 0$ exists $e = Ay$ exists too ("potentials exist")

Ohm's (or Hooke's) Law

- a graph becomes a *network* when we assign weights $c_1, \dots, c_m$ to the edges.
- Ohm's law specifies these weights based on physical principles.



Ohm's law $w_i = c_i e_i$ (edge flow)=(conductance)(potential difference)

$$\underbrace{\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix}}_{w} = \underbrace{\begin{bmatrix} c_1 & & & \\ & c_2 & & \\ & & \ddots & \\ & & & c_m \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_m \end{bmatrix}}_{e}$$

Equilibrium Equations: Strang Quartet

- *recipe for equilibrium equations:*

  (S1) express KVL via $\boldsymbol{e} = \boldsymbol{A}\boldsymbol{y}$

  (S1) express Ohm's law via $\boldsymbol{w} = \boldsymbol{C}\boldsymbol{e}$

  (S1) express KCL via $\boldsymbol{A}^T\boldsymbol{w} = \boldsymbol{x}$

  (S1) equilibrium equation: $\boldsymbol{A}^T\boldsymbol{C}\boldsymbol{A}\boldsymbol{y} = \boldsymbol{x}$

node potentials y

$\downarrow$

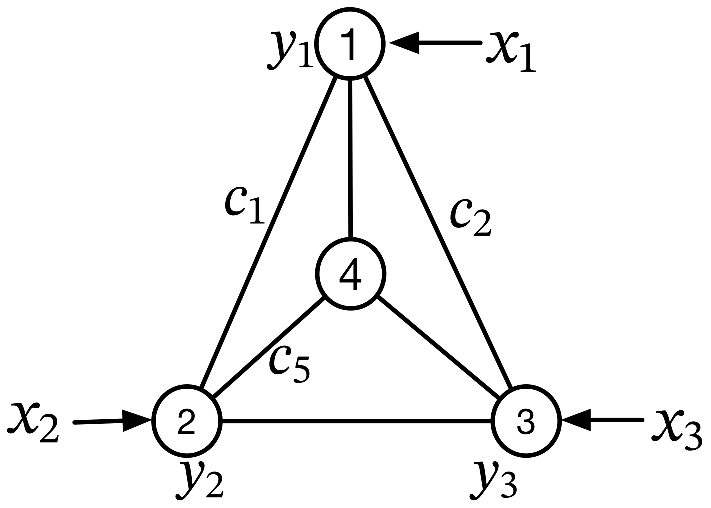
edge differences $\boldsymbol{e}$

$\downarrow$

edge flows $\boldsymbol{w}$

$\downarrow$

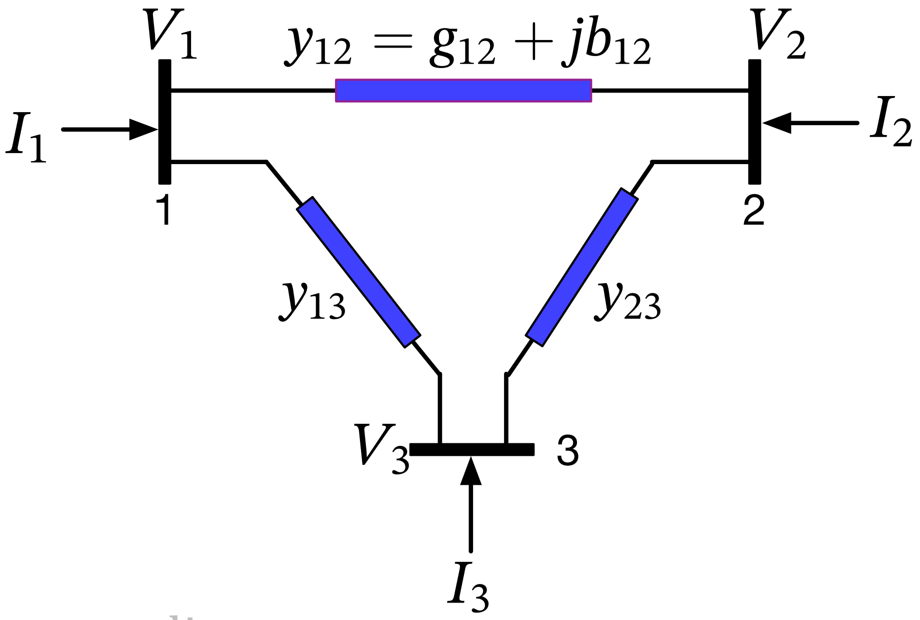
injected flows $\boldsymbol{x}$



Main point: equilibrium equation relate node potentials to injected flows

Example 1: Power Networks - AC formulation

Power network		Resistor network	
	Symbol	Quantity	Symbol
Complex injections	$\mathbf{I} \in \mathbb{C}$	Nodal current	$\mathbf{I}$ (or $\mathbf{x}$)
Complex edge flow	$\mathbf{w} \in \mathbb{C}$	Edge current flow	$\mathbf{w}$
Complex voltages	$\mathbf{V} \in \mathbb{C}$	Nodal voltages	$\mathbf{V}$ (or $\mathbf{y}$)
Line admittance	$y = g + ib$	Inverse edge resistance	$c = 1/r$



equilibrium equation (complex-current flow)

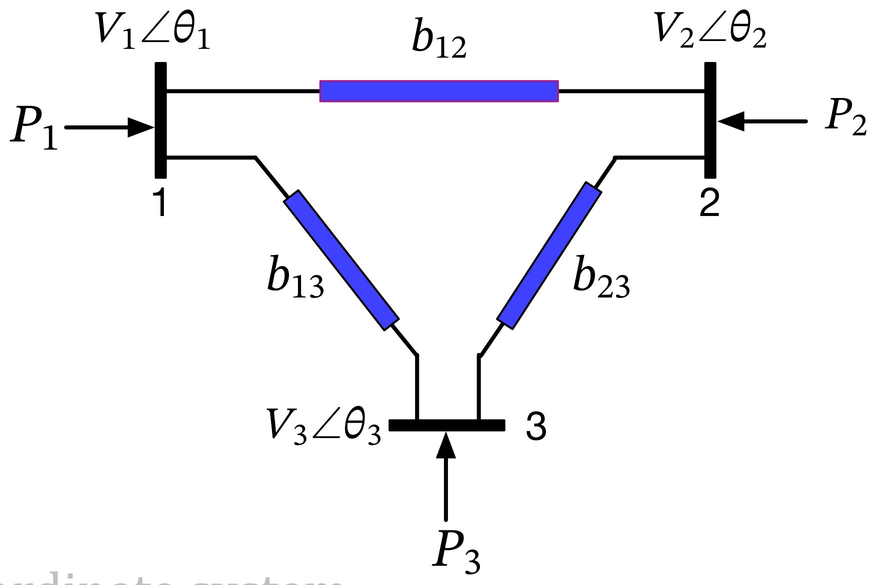
$$(A^T C A) V = \mathbf{I}$$

$$C = \begin{pmatrix} y_1 & & \\ & \ddots & \\ & & y_p \end{pmatrix}$$

cartesian-coordinate system

Example 2: Power Networks - DC formulation

dc approximation		Resistor network	
	Symbol	Quantity	Symbol
Power injections	$\mathbf{p}$	Nodal current	$\mathbf{I}$ (or $\mathbf{x}$)
Edge power flow	$\mathbf{w}$	Edge current flow	$\mathbf{w}$
Nodal phase angles	θ	Nodal voltages	$\mathbf{V}$ (or $\mathbf{y}$)
Line susceptances	b	Inverse edge resistance	$c = 1/r$



equilibrium equation (real-power flow)

$$A^T C A \theta = \mathbf{p}$$

$$C = \begin{pmatrix} b_1 & & \\ & \ddots & \\ & & b_p \end{pmatrix}$$

polar-coordinate system

Example 3: Mechanical Networks

Mechanical network		Resistor network	
	Symbol	Quantity	Symbol
Horizontal force	$\mathbf{f}$	Nodal current	$\mathbf{I}$ (or $\mathbf{x}$)
Restoring force	$\mathbf{w}$	Edge current flow	$\mathbf{w}$
Displacement	$\mathbf{d}$	Nodal voltages	$\mathbf{V}$ (or $\mathbf{y}$)
Stiffness	k	Inverse edge resistance	$c = 1/r$

equilibrium equation

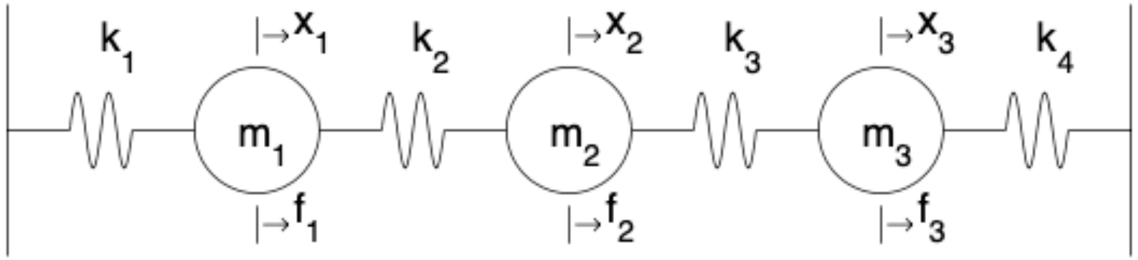


fig: fiber chain

$$(A^T C A) d = f$$

$$C = \begin{pmatrix} k_1 & & \\ & \ddots & \\ & & k_p \end{pmatrix}$$

applications: polymer networks, beam structures, hydraulic systems

Strang Quartet Extensions

physical systems:

- electrical networks with batteries and electronic parts (see [B1,B2, B5])
- water and gas networks (see [D3])
- transportation networks (see [D4])
- molecular and chemical reaction networks (see [B5])

beyond physical systems:

- leveling networks in GPS (see [B3]); and least squares problems (see [B3])
- ranking of football teams (see [B2]) (maybe cricket?)

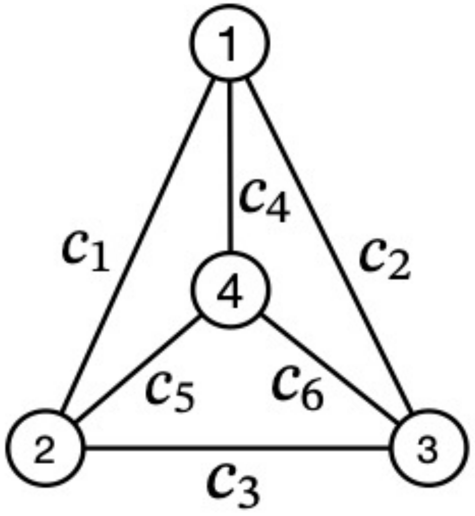
Strang Quartet Extensions

dynamical systems:

❧ replacing x with dx/dt and d^2x/dt^2 gives first- and second-order ODE

❧ replacing finite dim. vectors x with continuous functions gives PDEs and other continuous forms of Kirchoff's laws (see [P1`])

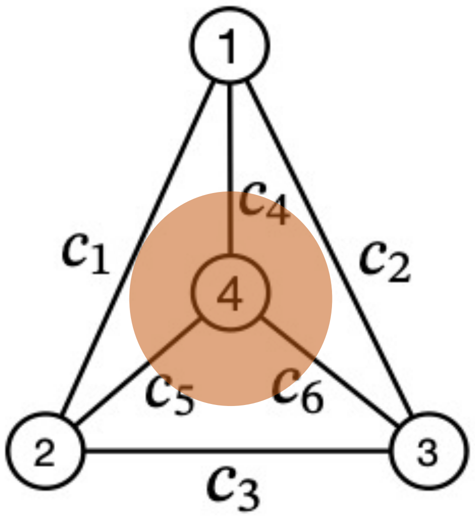
The Laplacian Matrix



$$\underbrace{\begin{bmatrix} c_1+c_2+c_4 & -c_1 & -c_2 & -c_4 \\ -c_1 & c_1+c_3+c_5 & -c_3 & -c_5 \\ -c_2 & -c_3 & c_2+c_3+c_6 & -c_6 \\ -c_4 & -c_5 & -c_6 & c_4+c_5+c_6 \end{bmatrix}}_{L=A^TCA}$$

- Laplacian $L \in \mathbb{R}^{p \times p}$ gives information about edge connections and weights
- our Laplacian is not invertible, is positive semi-definite, and is undirected (exercise)
- applications: *consensus, network science, graph signal processing, and Laplacian systems (Lx=b)*
- reduced L is obtained by deleting a row/column pair (or by deleting a column of A)

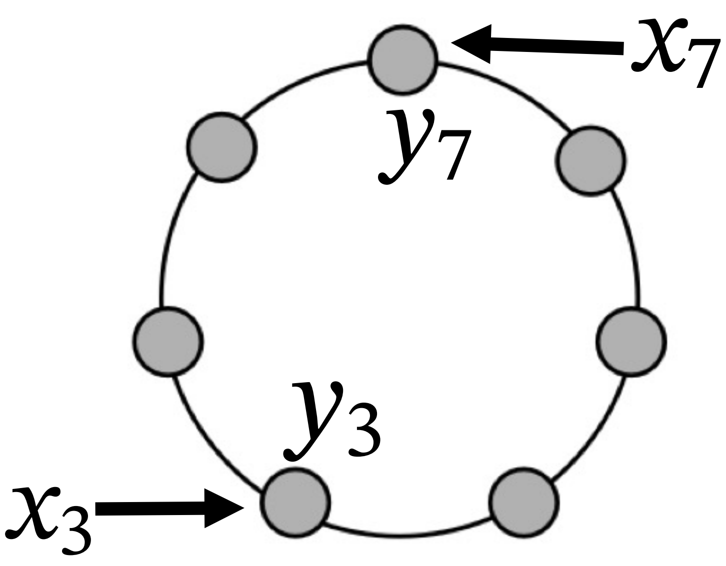
Reduced-order Laplacian Matrix



$$\underbrace{\begin{bmatrix} c_1+c_2+c_4 & -c_1 & -c_2 & -c_4 \\ -c_1 & c_1+c_3+c_5 & -c_3 & -c_5 \\ -c_2 & -c_3 & c_2+c_3+c_6 & -c_6 \\ -c_4 & -c_5 & -c_6 & c_4+c_5+c_6 \end{bmatrix}}_{L=A^TCA}$$

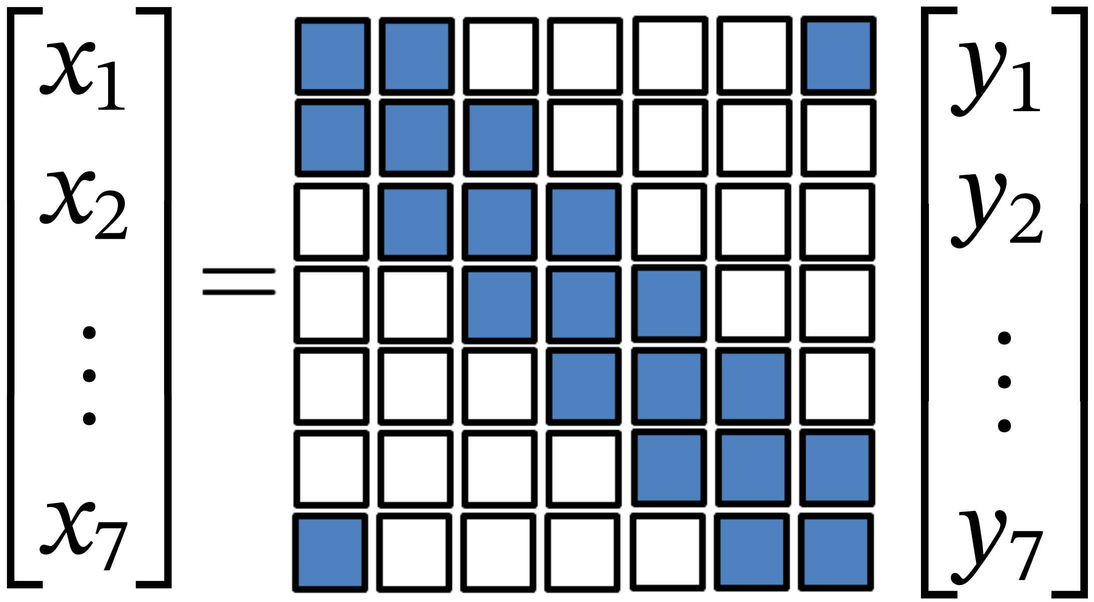
- Laplacian $L \in \mathbb{R}^{p\times p}$ gives information about edge connections and weights
- our Laplacian is not invertible, is positive semi-definite, and is undirected (exercise)
- applications: *consensus, network science, graph signal processing, and Laplacian systems (Lx=b)*
- reduced L is obtained by deleting a row/column pair (or by deleting a column of A)

Network Structure = Laplacian’s Sparsity Pattern



infrastructure network

sparsity (zero & non-zero) of L
captures network connections

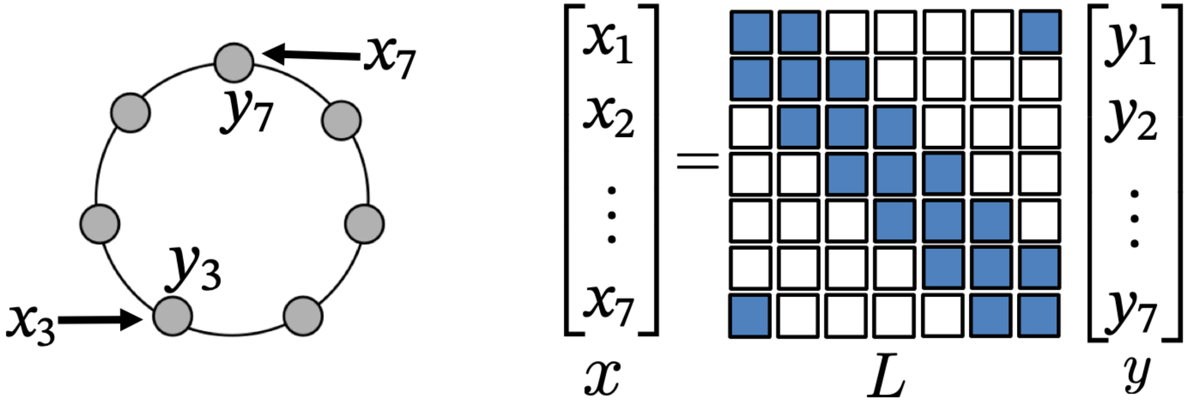


x nodal injections

L network Laplacian

y node potentials

Structure Learning Problem Setup



$$x,\ y \in \mathbb{R}^{p\times p}$$

$$L \in \mathbb{R}^{p\times p}$$

problem: learn sparsity of L from data $x(k), y(k)$, with $k = 1, \ldots, K$

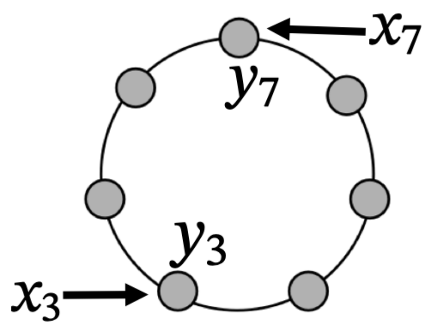
assumptions on sensors

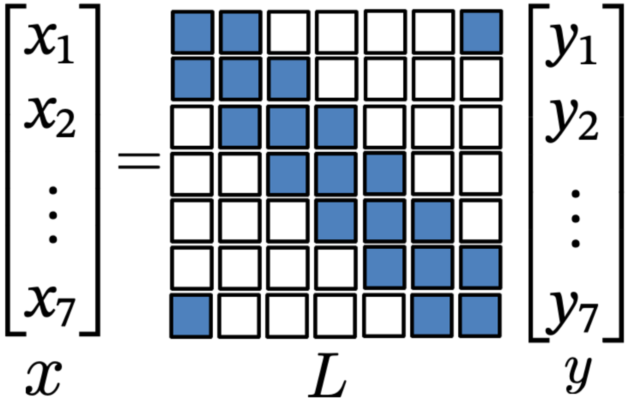
full coverage $\Leftrightarrow$ data from all nodes

partial coverage $\Leftrightarrow$ data from a subset of nodes (sub-vectors of x, y)

Note: all results hold for complex-valued vectors and matrices

Structure Learning Problem Setup





$$x,\ y \in \mathbb{R}^{p\times p}$$

$$L \in \mathbb{R}^{p\times p}$$

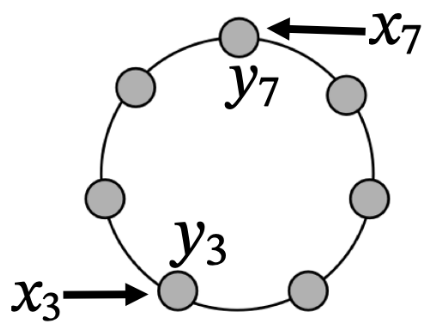
problem: learn sparsity of L from *full or partial coverage sensors*

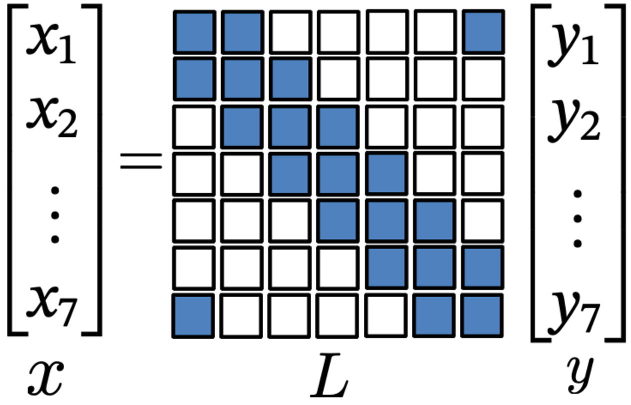
assumptions on data

linear models $\Leftrightarrow$ access $x(k)$ and $y(k)$; with some additive noise

covariance models $\Leftrightarrow$ access $y(k)$; with prior on the covariance of $x(k)$

Structure Learning Problem Setup





$$x,\ y \in \mathbb{R}^{p \times p}$$

$$L \in \mathbb{R}^{p \times p}$$

problem: learn sparsity of L from *full or partial coverage sensors*

data regimes

classical (low-dimensional) $\iff$ $K \gg p$ (more samples than nodes)

modern (high-dimensional) $\iff$ $K \ll p$ (less samples than nodes)

Linear Models (full-coverage)

- model with additive noise: $x(k) = Ly(k) + \varepsilon(k)$

- matrix-valued equation

$$\underbrace{\begin{bmatrix} x(1) & \cdots & x(K) \end{bmatrix}}_{X} = L \underbrace{\begin{bmatrix} y(1) & \cdots & y(K) \end{bmatrix}}_{Y} + \underbrace{\begin{bmatrix} \varepsilon(1) & \cdots & \varepsilon(K) \end{bmatrix}}_{E}$$

- least squares (LS): $X \in \mathbb{R}^{p \times K}$ full row-rank

$$\hat{L}_{\text{ols}} = \arg\min_{L \in \mathbb{R}^{p \times p}} \|LY - X\|_F^2 = XY^T(YY^T)^{-1}$$

exercise!

  - **pros:** (i) simple; (ii) full-order L
  - **cons:** (i) classical regime ($K \geq p$); (ii) neglects symmetry and $L1 = 0$

Linear Models (full-coverage)

- vectorized equilibrium equation:

$$X = LY + E$$ matrix-valued

$$\mathrm{Vec}(X) = H(Y)\,\mathrm{Ve}(L) + \mathrm{Vec}(E)$$ vector-valued

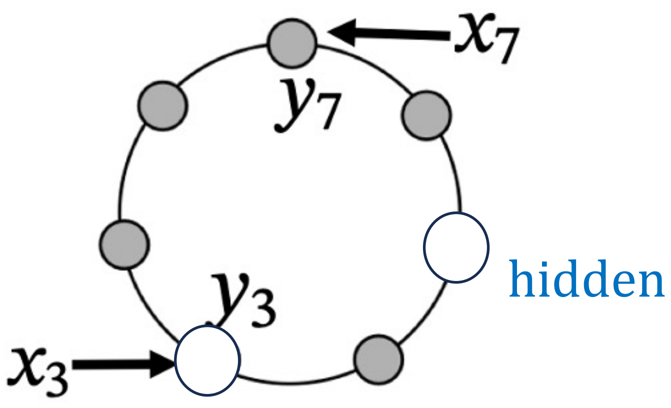
- example (full details at the end of the slides)

$$L = \begin{bmatrix} a_1 + a_2 & -a_1 & -a_2 \\ -a_1 & a_1 + a_3 & -a_3 \\ -a_2 & -a_3 & a_2 + a_3 \end{bmatrix} \quad \mathrm{vech}(L) = \begin{bmatrix} a_1 + a_2 \\ -a_1 \\ -a_2 \\ a_3 + a_1 \\ -a_3 \\ a_2 + a_3 \end{bmatrix} \quad \mathrm{ve}(L) = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Linear Models (partial coverage)

- partition the equilibrium equation:

$$\underbrace{\begin{bmatrix} x_O(k) \\ x_H(k) \end{bmatrix}}_{x(k)} = \underbrace{\begin{bmatrix} L_{OO} & L_{OH} \\ L_{HO} & L_{HH} \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} y_O(k) \\ y_H(k) \end{bmatrix}}_{y(k)}$$



- set the flow injection $x_H(k) = 0$ (valid assumption in power networks)

$$\begin{bmatrix} x_O(k) \\ 0 \end{bmatrix} = \begin{bmatrix} L_{OO} & L_{OH} \\ L_{HO} & L_{HH} \end{bmatrix} \begin{bmatrix} y_O(k) \\ y_H(k) \end{bmatrix}$$

- L_{HH} is invertible (connectivity of L guarantees invertibility)

Linear Models (partial-coverage)

- Kron-reduction (aka Schur's complement):

$$\begin{bmatrix} x_O(k) \\ 0 \end{bmatrix} = \begin{bmatrix} L_{OO} & L_{OH} \\ L_{HO} & L_{HH} \end{bmatrix} \begin{bmatrix} y_O(k) \\ y_H(k) \end{bmatrix} \implies x_O = \underbrace{\left(L_{OO} - L_{OH} L_{HH}^{-1} L_{HO}\right)}_{\bar{L}} y_O$$

- solve the second block for *y_H* and substitute it in the first

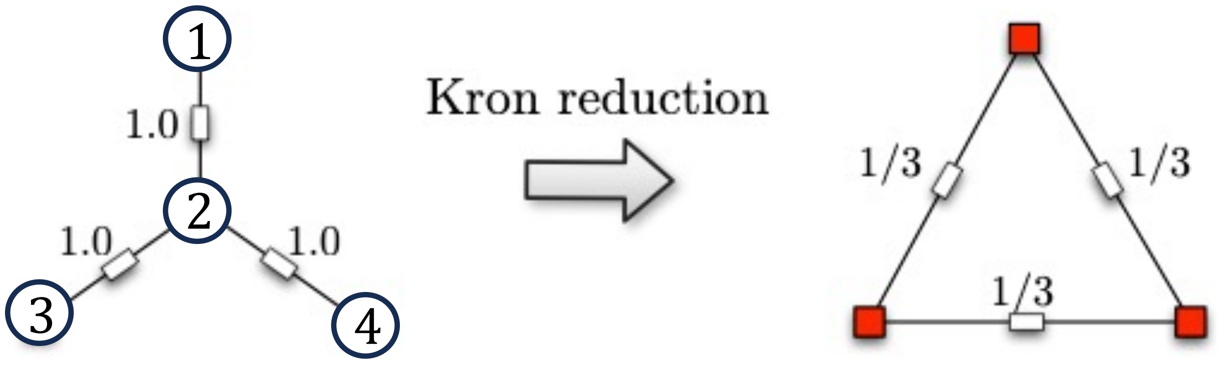


fig source: web

Linear Models (partial-coverage)

- Kron reduction for the figure on the previous slide:

$$L = \begin{bmatrix} L_{OO} & L_{OH} \\ L_{HO} & L_{HH} \end{bmatrix} = \left[\begin{array}{ccc|c} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 1 & 0 \\ \hline 0 & -1 & 0 & 1 \end{array}\right]$$

$$\overline{L} = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} \end{bmatrix}$$

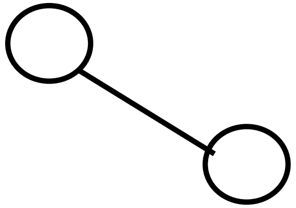
Linear Models (partial coverage)

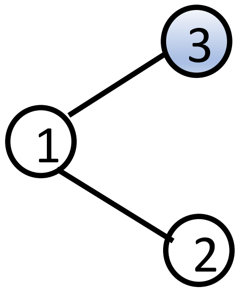
- can we estimate L (i.e., sparsity pattern) from $\overline{L}$? no

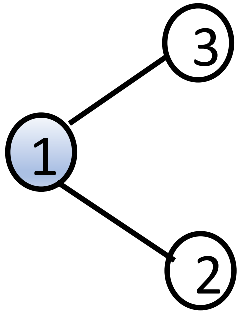
- Kron reduction is not injective

$$\overline{L} = \begin{bmatrix} l & -l \\ -l & l \end{bmatrix} \qquad L_1 = \left[\begin{array}{cc|c} l + l' & -l & -l' \\ -l & l & 0 \\ \hline -l' & 0 & l' \end{array}\right] \qquad L_2 = \left[\begin{array}{cc|c} l_1 & 0 & -l_1 \\ 0 & l_2 & -l_2 \\ \hline -l_1 & -l_2 & l_1 + L_2 \end{array}\right]$$

$$l_1 l_2 / (l_1 + l_2) = l$$







- for tree networks, it is possible under certain assumptions (see [P3])

Wrap Up

- equilibrium equations allow us to model
  - infrastructure networks (e.g., power, water, gas)
  - non-infrastructure networks (e.g., metabolic, chemical, leveling)
- network structure is given by the off-diagonals of the Laplacian
- learning using linear models need both injections and potentials
- partial-coverage setting is more challenging than the full coverage

To learn more…

Contact: rangulur@asu.edu https://rajanguluri.github.io (lecture notes: coming soon)

Books and notes:

[B1] G. Strang (2019). Linear algebra and learning from data, Wellelsey Publishers

[B2] G. Strang (2007). Computational science and engineering, Wellelsey Publishers

[B3] G. Strang (1997). Linear algebra, geodesy, and GPS, SIAM

[B4] E. B. Curtis and J.A. Morrow (2000). Inverse problems in electrical engineering, World Scientific Press

[B5] S. J. Cox (2018). Linear algebra *in Situ*, CAAM 335, lecture notes, Rice University (link: click)

[B6] N. K. Vishnoi (203). Lx=b, Foundations and Trends in Theoretical Computer Science, vol 8, no 1-2

[B7] S. Low (2024). Power system analysis: a mathematical approach, lecture notes, Caltech (link: click)

Dissertations:

[D1] *M. Babakmehr (2017) Compressive power systems, Colorado School of Mines (link: click)

[D2] M. Bariya (2022) Applications of time synchronized measurements in the electric grid, UC Berkely (link: click)

[D3] V. Singh (2018) Optimal operation of water and power distribution networks, Virgina Tech (link: click)

[D4] F. Seccamonte (2023) Bilevel optimization in learning and control with applications to network flow and estimation, UCSB (link: click)

*on linear model structure learning

To learn more…

Contact: rangulur@asu.edu https://rajanguluri.github.io (lecture notes: coming soon)

Selective Papers:

[P1] G. Strang (1988) A framework for equilibrium equations, SIAM Review, 30(2)

[P2] F. Kaiser and D. Witthaut (1988) Topological theory of resilience and failure spreading in flow networks, Physical review research, 3(2)

[P3] *Y. Yuan et. al. (2022) Inverse power flow problem, IEEE Trans on Control of Network Systems, 10(1)

[P4] *O. Ardakanian et. al. (2019) On identification of distribution Grids, IEEE Trans on Control of Network Systems, 6(3)

[P5] D. Deka et. al. (2023) Learning distribution grid topologies, IEEE Trans on Smart Grid, 15(1)

[P6] G. Cavraro et. al. (2021) Learning power grid topologies, in Advanced Data Analytics in Power Systems, Cambridge (link: click)

[P7] F. Dorfler and F. Bullo (2012) Kron reduction of graphs with applications to electrical networks, IEEE Trans on Circuits on Systems, 60(1)

[P8] F. Seccamonte et.al. (2023) Inference of infrastructure network flows via physic-inspired implicit neural networks, IEEE CCTA

[P9] K. D. Smith et.al. (2022) Physics-informed implicit representations of equilibrium network flows, NeurIPS, 35

[P10] A. Silva et.al. (2021) Combining physics and machine learning for network flow estimation, ICLR, 2020

[P11] S. Pappu et.al. (2021) Predicting unknown directed links of conserved networks from flow data, Journal of Complex Networks, 9(6)

[P12] S. Pappu et.al. (2022) Verification and rectification of error in topology of conserved networks, IFAC, 2022

*on linear model structure learning

Back up slides

Laplacian Estimation: Full Coverage

- **ordinary least squares (OLS):** Matrices $X, Y \in \mathbb{R}^{p\times K}$ are known. Let $Y \in \mathbb{R}^{p\times K}$ be full row-rank matrix. Then,

$$\hat{L}_{\text{ols}} = \underset{L \in \mathbb{R}^{p\times p}}{\arg\min} \|LY - X\|_F^2$$
$$= XY^T(YY^T)^{-1}$$

- *proof*: write $||A||_F^2 = \mathrm{Tr}(AA^T)$ and set the derivative to zero
- works in classical regime ($K \geq p$)
- neglects: symmetry ($L = L^T$) and Laplacianity ($L\mathbb{1} = 0$)

Laplacian Estimation: Full Coverage

- **constrained least squares (CLS):**

$$\hat{L}_{\text{cls}} = \underset{L=L^T \in \mathbb{R}^{p\times p}}{\arg\min} \|LY - X\|_F^2$$

$$\text{subject to } L1 = 0$$

- convert the CLS problem into OLS via the vectorization trick:

$$\begin{aligned}\mathrm{Vec}(X) &= \mathrm{Vec}(LY) + \mathrm{Vec}(E)\\ &= (Y^T \otimes I_p)\,\mathrm{Vec}(L) + \mathrm{Vec}(E)\end{aligned}$$

- $\mathrm{Vec}(L) \in \mathbb{R}^{p^2}$ contains $\frac{p(p-1)}{2} + p$ redundant elements

Laplacian Estimation: Full Coverage

- there exists a Duplication and Transformation matrices D and T:

$$\mathrm{Vec}(L) = D \underbrace{\mathrm{Vech}(L)}_{\mathbb{R}^{[p(p-1)/2+p]}} = TD \underbrace{\mathrm{Ve}(L)}_{\mathbb{R}^{p(p-1)/2}}$$

example:

$$L = \begin{bmatrix} a_1 + a_2 & -a_1 & -a_2 \\ -a_1 & a_1 + a_3 & -a_3 \\ -a_2 & -a_3 & a_2 + a_3 \end{bmatrix} \quad \mathrm{vech}(L) = \begin{bmatrix} a_1 + a_2 \\ -a_1 \\ -a_2 \\ a_3 + a_1 \\ -a_3 \\ a_2 + a_3 \end{bmatrix} \quad \mathrm{ve}(L) = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Laplacian Estimation: Full Coverage

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{9\times 6}$$

(duplication matrix)

$$T = \begin{bmatrix} 1 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & -1 \\ 0 & 1 & 1 \end{bmatrix}_{6\times 3}$$

(transformation matrix)