

## TUTORIAL II: NETWORK THEORY AND POWER FLOW

---

### PROBLEM II.1 (PROGRAMMING) – NETWORK THEORY BASICS 🧑‍🤝‍🧑

Consider the simple network shown in Figure 1. Calculate in Python or by hand:

- (a) Compile the *nodes list* and the *edge list*.

**Remark:** While graph-theoretically both lists are unordered sets, let's agree on an ordering now which can serve as basis for the matrices in the following exercises: we sort everything in ascending numerical order, i.e. node 1 before node 2 and edge (1,2) before (1,4) before (2,3).

- (b) Determine the *order* and the *size* of the network.
- (c) Compute the *adjacency matrix*  $A$  and check that it is symmetric.
- (d) Find the *degree*  $k_n$  of each node  $n$  and compute the *average degree* of the network.
- (e) Determine the *incidence matrix*  $K$  by assuming the links are always directed from smaller-numbered node to larger-numbered node, i.e. from node 2 to node 3, instead of from 3 to 2.
- (f) Compute the *Laplacian*  $L$  of the network using  $k_n$  and  $A$ . Remember that the Laplacian can also be computed as  $L = KK^T$  and check that the two definitions agree.
- (g) Find the *diameter* of the network by looking at Figure 1.

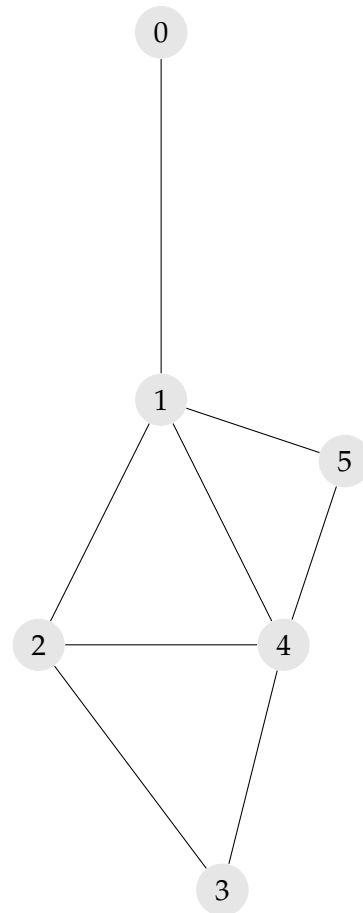


Figure 1: Simple Network

## PROBLEM II.2 (PROGRAMMING) – LINEAR POWER FLOW 🐼

If you map the nodes to countries like 0=DK, 1=DE, 2=CH, 3=IT, 4=AT, 5=CZ the network in Figure 2 represents a small part of the European electricity network (albeit very simplified). You can find the *power imbalance* time series for the six countries for January 2017 in hourly MW in the file `imbalance.csv`. They have been derived from physical flows as published by ENTSO-E.<sup>1</sup>

The linear power flow is given by

$$p_i = \sum_j \tilde{L}_{i,j} \theta_j \quad \text{and} \quad f_l = \frac{1}{x_l} \sum_i K_{i,l} \theta_i, \quad \text{where} \quad \tilde{L}_{i,j} = \sum_l K_{i,l} \frac{1}{x_l} K_{j,l} \quad (1)$$

is the weighted Laplacian. For simplicity, we assume identity reactance on all links  $x_l = 1$ .

- (a) Compute the *voltage angles*  $\theta_j$  and *flows*  $f_l$  for the first hour in the dataset with the convention of  $\theta_0 = 0$ ; i.e. the slack bus is at node 0.

**Remark:** Linear equation systems are solved efficiently using `numpy.linalg.solve`.

- (b) Determine the average flow on each link for 01-2017 and draw it as a directed network.

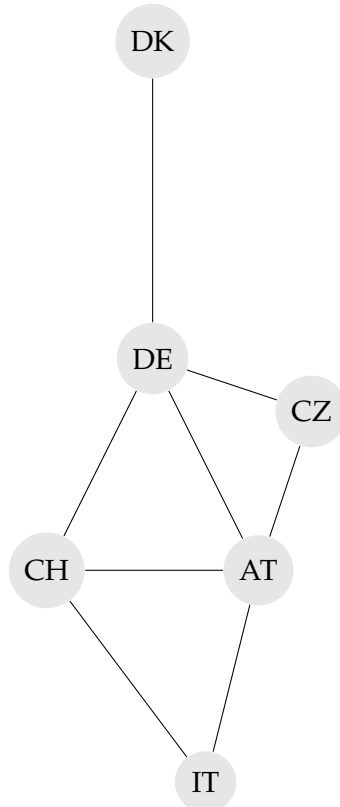


Figure 2: Simple Network

<sup>1</sup><https://transparency.entsoe.eu/transmission-domain/physicalFlow/show>