Welcome.

Everyone:

- Pull the updates from the course GitHub repo:
  - `cd <46120-PiWE repo>`
  - `git pull upstream main` ← you might have “upstream2” instead

Physical students:

- Sit with your P0 Team.
- Turn off laptop volume (mute). ←IMPORTANT!
- Log into the Zoom meeting.
  - Link is on Learn webpage, under “PiWE Links” on main page.
- Microphone muted. Camera off.

46120: Scientific Programming for Wind Energy

Turbie

Jenni Rinker

Agenda for today.

- Pull new course material ✔️

- Round robin.
- Turbie.

- Begin teamwork on Week 3 homework.
  - Form CodeCamp teams! Deadline is **Monday Feb. 24 23.59**. “EM” grade if not properly signed up for a team.

Round robin

Share solutions with your peers and give feedback.

Time to review and collaborate.

- 1 round of 30 minutes.
- 5 minutes: chaos.
- 25 minutes: present/discuss homework.
  - Functions AND tests! Discuss also the numpy/matplotlib tutorials/exercises.
  - Team A screenshares & presents their solutions. Teams B & C provides feedback.
  - Switch which group presents/provides feedback.
- Afterwards: plenum discussion.
  - Be ready with questions!

- Teams in breakout rooms (BORs):

	P0 Teams
BOR 1	12, 24, 15, 7
BOR 2	11, 18, 13
BOR 3	17, 0, 6
BOR 4	21, 16, 1
BOR 5	9, 22, 8
BOR 6	14, 5, 19
BOR 7	2, 20, 4
BOR 8	23, 10, 3

Code used to generate:

```
import random
groups = list(range(25))
print(random.shuffle(groups))
```

[and then some manual entry]

Notes in plenum.

- Add here.

- error in 3a tutorial, Javascript error in matplotlib tutorial
  cell %matplotlib notebook can be deleted

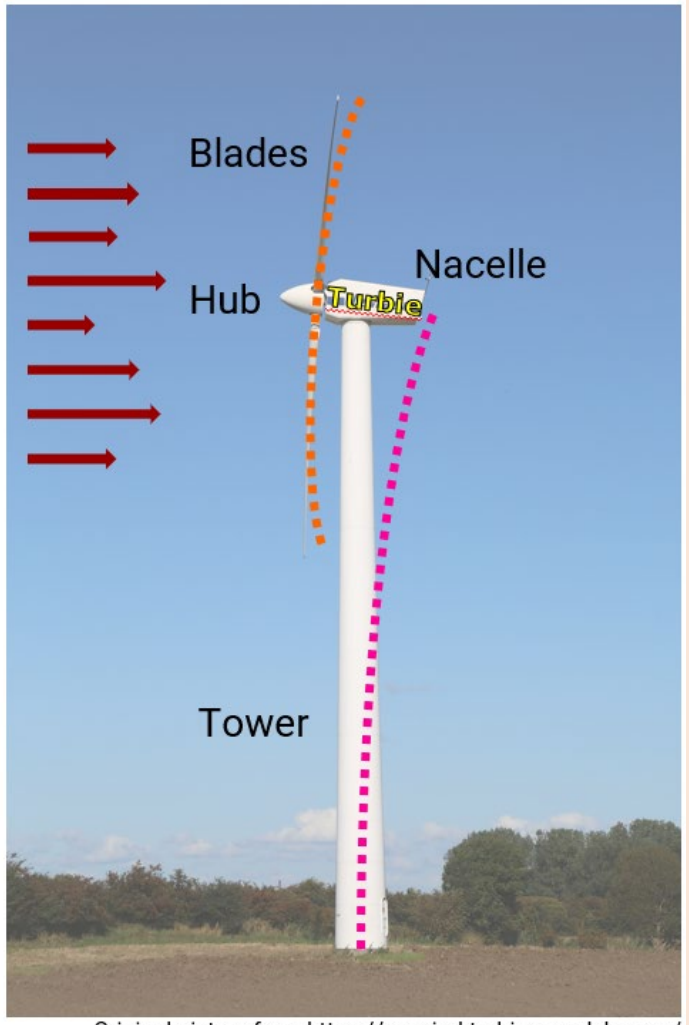
Question: can we instead design the function to return the string, so we dont' have to capture the output?
yes, depends on what you want the function to do

Turbie

A beautiful windy girl.

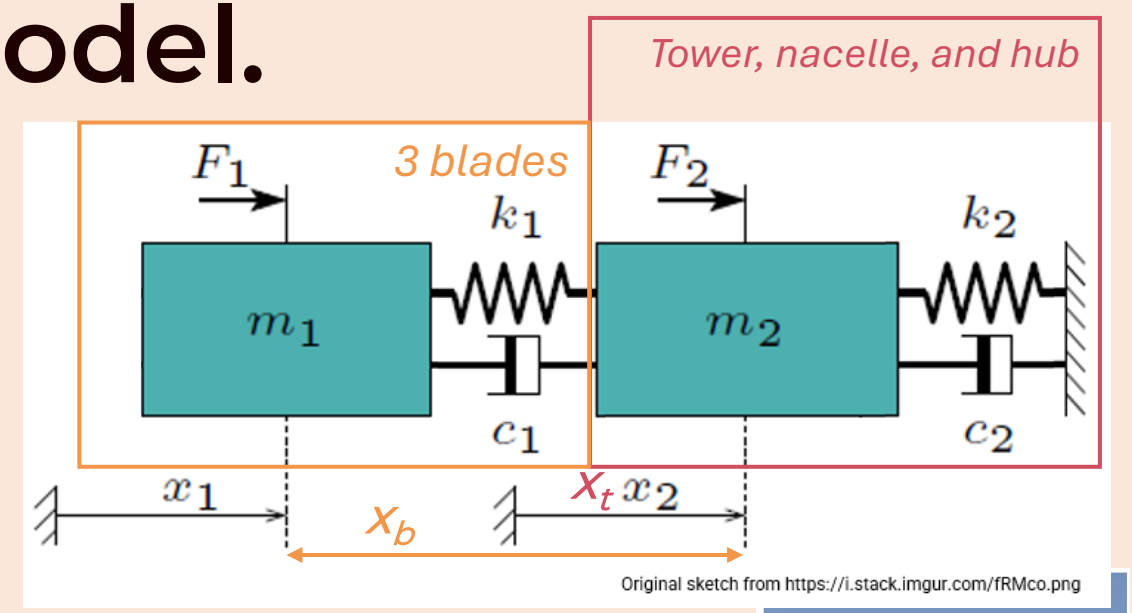
Project: simulate response of a wind turbine to turbulence.

- Wind turbine model with 2 flexible DOF.
  - Blade collective flap deflection
  - Tower fore-aft deflection
- Time-varying wind loads applied on blades cause time-varying response in the 2 DOFs.
- Will model and simulate this dynamical system.
  - *(Using simplified physics.)*



Original picture from https://en.wind-turbine-models.com/

Simple 2DOF model.



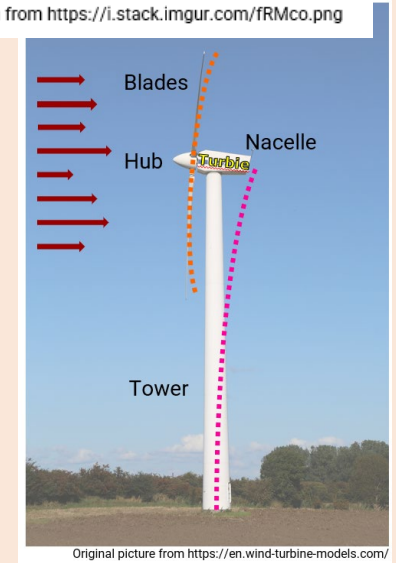
Original sketch from https://i.stack.imgur.com/fRMco.png

- 2 DoF mass-spring-damper

- m_2 is the combined mass of hub, nacelle and tower
- x_2 (x_t) is the towertop deflection (relative to ground)

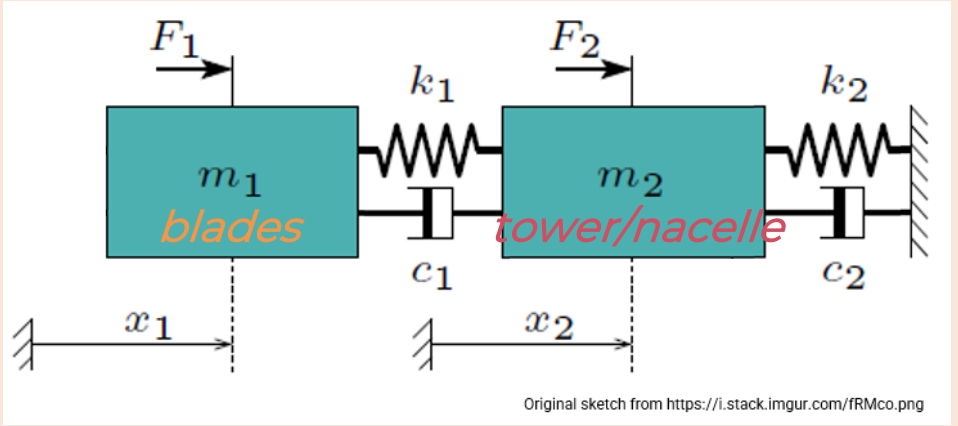
- m_1 is the combined mass of 3 blades
- x_1 is the absolute blade tip deflection relative to ground

- $x_b = x_1 - x_2$ is the blade tip deflection relative to tower



Original picture from https://en.wind-turbine-models.com/

Equations of motion and parameters.



- Equations of motion for this 2DOF system:

$$\boldsymbol{M\ddot{x} + C\dot{x} + Kx = F}$$

with

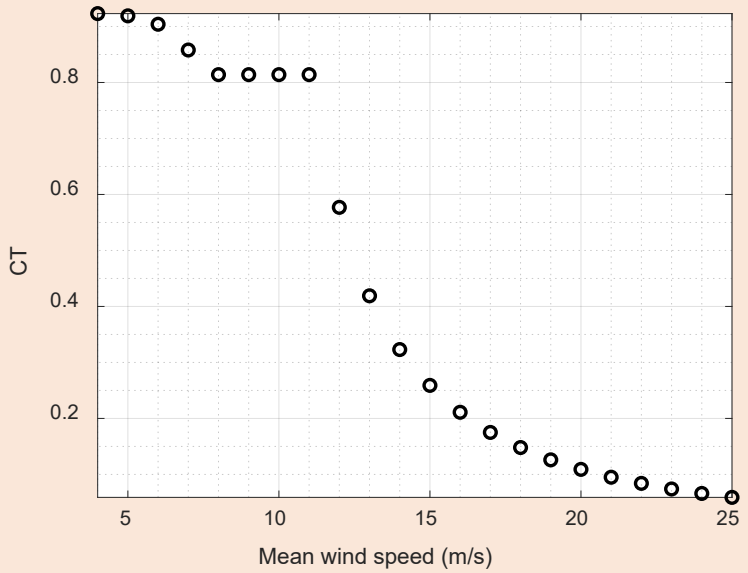
$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \boldsymbol{M} = \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \qquad \boldsymbol{C} = \begin{bmatrix} c_1 & -c_1 \\ -c_1 & c_1 + c_2 \end{bmatrix} \qquad \boldsymbol{K} = \begin{bmatrix} k_1 & -k_1 \\ -k_1 & k_1 + k_2 \end{bmatrix} \qquad \boldsymbol{F} = \begin{bmatrix} F_1 \\ 0 \end{bmatrix}$$

aerodynamic forcing!

- All parameters given in turbie_parameters.txt.
  - File in codecamp team repo (will clone soon), under data/ folder.

(Overly) simple model of aerodynamic forcing.

- Assume rotor thrust coefficient (C_T) constant for a 10-minute simulation.
  - But it *is* a function of mean wind speed!



Wind force on the rotor is modelled as

$$f_{aero}(t) = \frac{1}{2}\rho A_r C_T[V(t) - \dot{x}_1(t)]|V(t) - \dot{x}_1(t)|$$

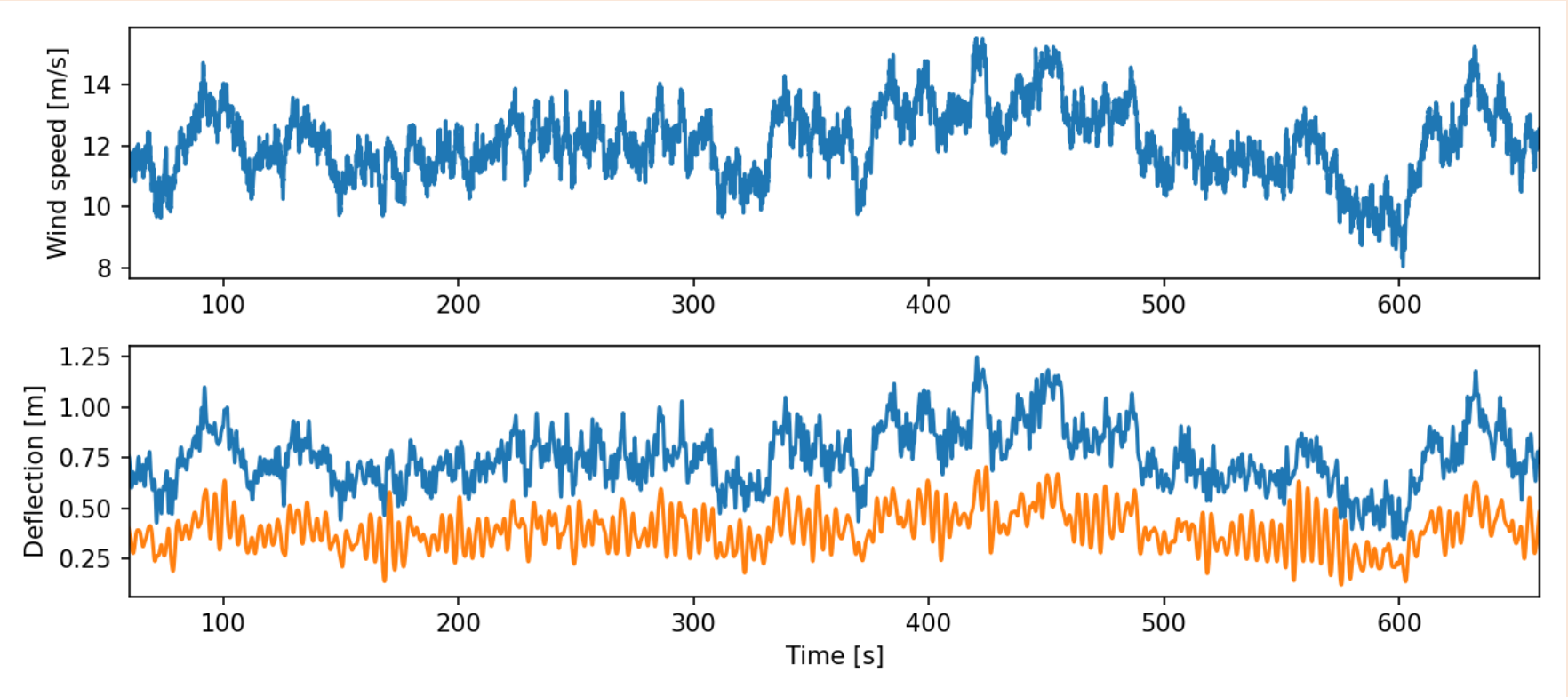
with

ρ	air density	$V(t)$	wind speed at hub
A_r	rotor area	$\dot{x}_1(t)$	blade velocity
$C_T(\bar{V})$	thrust coefficient		

- All parameters given in turbie_parameters.txt and CT.txt, in data/ folder on codecamp team repo.

What your code will do by the end.

- By the end of the CodeCamp module you will generate results like this



AND analyze statistics as a function of wind speed!

Overview of the CodeCamp project.

	Tasks	Notes
Week 3	Writing CodeCamp functions, Part 1	Specifically defined
Week 4	Writing CodeCamp functions, Part 2	Specifically defined
Week 5	Diagramming code and introduction to CodeCamp “final” project	Teams design the solution
Week 6	Presentations of CodeCamp “final” project	Must pass CodeCamp project to submit a final project
Week 7	Turn CodeCamp into a package	Reusing Turbie to explain how installing packages works

Questions?

Homework for this week

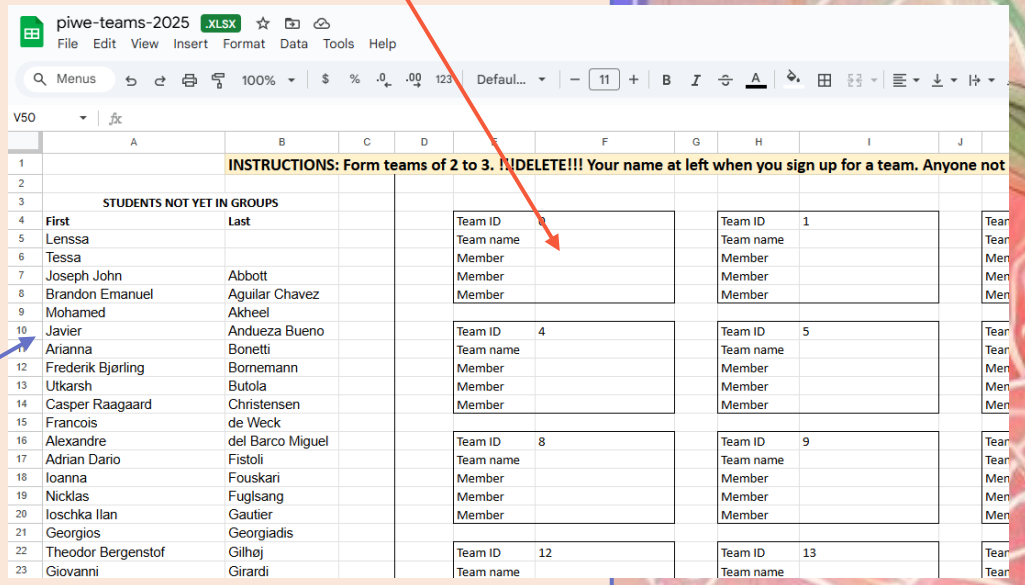
Go forth and meet your destiny.

First things first: group formation.

- Groups of **2 to 3 students** for the CodeCamp project.
- Process to form groups written in this week's homework.
- **VERY IMPORTANT for proper sign-up**:
  1. Name deleted in left column in sign-up sheet AND FULL name listed under a team.
  2. You must have joined your team in the CodeCamp GitHub assignment.
- Deadline to sign up is **Monday Feb. 24, 23:59**.
  - Anyone listed in left pane or not on GitHub assignment by this deadline will get an "EM" (no show) grade, unless they drop the class.

add full name here!

delete name from here!



*Remember, you're expected to work **about 6 hours outside of class**. Schedule accordingly.*

Homework.

- Detailed on the course GitHub repo.
  - **Short summary**: form groups, write functions to load data from text files, plot time series, and create numpy arrays of system matrices.
- We'll open BORs in a minute.
  - Each team enters their BOR (same as your team ID). Perhaps find a physical spot outside the auditorium.
- Complete **Part 1** of the weekly assignment in class, then move on as agreed with your team.
- **To get help during class**: Post in Slack / #debugging if you want a TA to enter your BOR or come find your group.

Any questions?