ME 2016 A- Computing Techniques
Fall 2018

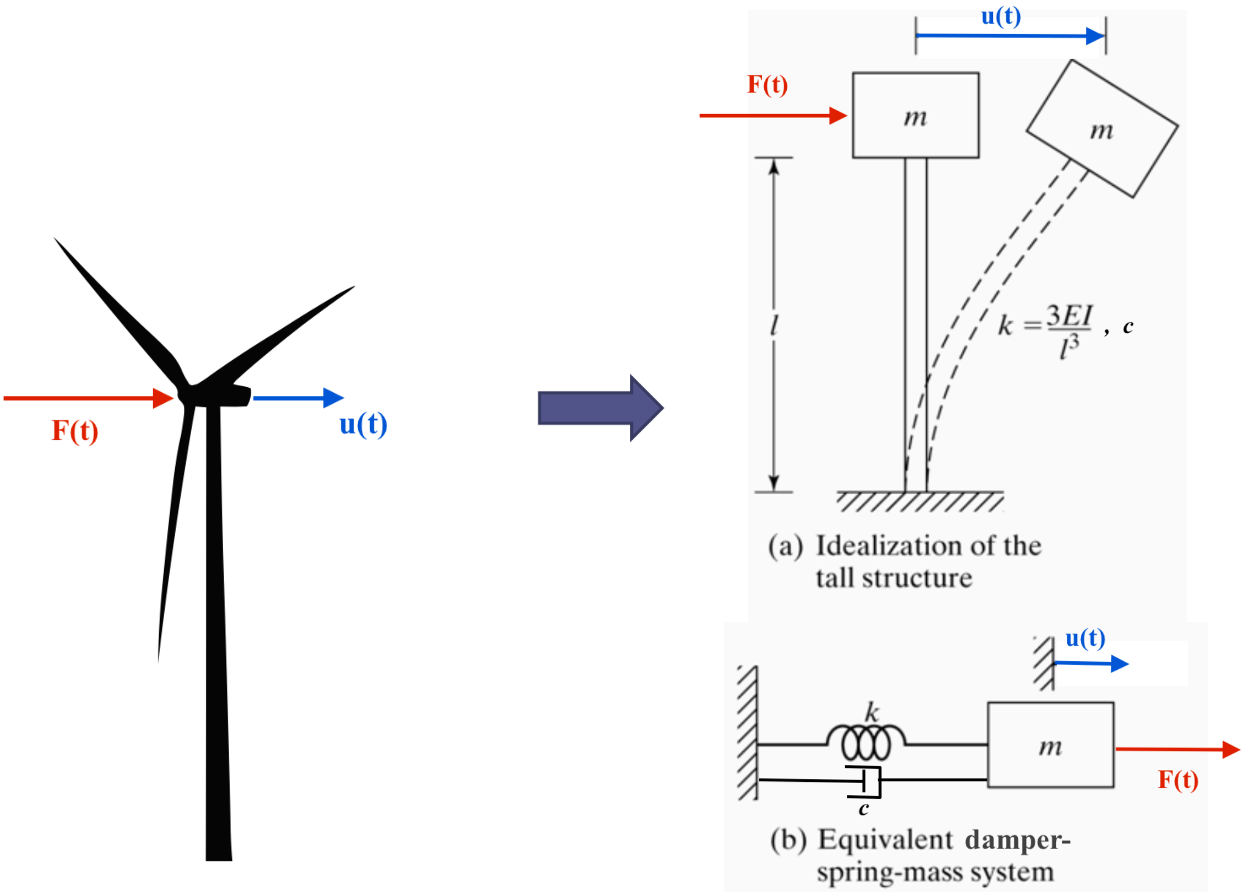
Computer Project 3
Due Tuesday, November 20 at 2:55 pm (5 minutes before class time)

Ordinary Differential Equations: Initial Value Problem

Your growing reputation as an alternative energy expert has led to a new job on the West coast, where you are a project manager in the Wind Energy Division of Southern California Edison.

<u>Problem description:</u>

Your team needs to study the vibration of a typical wind turbine in response to external excitations, such as those caused by an earthquake or by periodic variations in the wind flow. To that end, they are using a simple one-degree-of-freedom spring-damper-mass system to model the tower vibration, as shown below.



(a) Idealization of the tall structure

(b) Equivalent damper-spring-mass system

The equation of motion from Newton's second law corresponding to this model is

$$m\ddot{u} + c\dot{u} + ku = F(t)$$

where the dots mean derivatives with respect to time, $u(t)$ is the displacement of the top of the tower, $F(t)$ is the external force acting on the turbine, m is the mass of the turbine, k is the effective stiffness of the tower, and c is a coefficient representing structural damping in the system. The force acting on the turbine is assumed to be periodic and is represented by a sinusoidal function of the form $F(t) = A\sin(2\pi f t)$, where A is the force amplitude (N) and f is the frequency (Hz).

The goal of this project is to determine how the tower shakes by solving the differential equation of motion for a scaled-down model with the following parameter values: m = 9.0x10^3 kg, k = 1.05x10^6 N/m, c = 9.0x10^3 N.s/m and A = 3.0x10^4 N.

MATLAB programming:

1) Write a **completely generic** function to implement both Euler's and Heun's methods (you can use your Euler.m function from HW0 as a template). Your function should have the form:

```
function [T,Y]= my_ODE_solver(odefun,tspan,y0,dt,method)
```

where:

- `odefun` is a function handle that defines the ODE system to solve
- `tspan` is the integration interval vector [t0 tf]
- `y0` is a ***row*** vector of initial conditions
- `dt` is the step size (s)
- `method` is a string to specify that either Euler or Heun will be used
- `T` is a ***column*** vector of times at which solutions are computed: [0 dt 2dt 3dt…tf]^T
- `Y` is the solution matrix

This generic function should be able to handle systems of equations of ANY size (not just 2, as in this project). Its solution matrix `Y` should be structured like the one outputted by ODE45 (see the MATLAB help for ODE45). Please note that this means that you will have to convert the computed slopes (obtained from `odefun`) from column to row vectors in `my_ODE_solver`.
Hint: It is highly recommended that you do HW5 before starting coding this function: doing similar but simpler calculations by hand will help you understand how `my_ODE_solver` should work.

2) Write a function (specific to this particular problem) of the form

```
function shakemax = towershake(f,method,output_logical)
```

to solve the equation of motion for u. In this function:

- `f` is the forcing frequency (Hz)
- `method` is a string to specify that Euler, Heun or ODE45 will be used
- `shakemax` is the maximum value of the displacement (m) for later times (i.e. towards the end of the integration time window)
- `output_logical` is either the string "true" or "false" to determine if plot(s) and results will be displayed; if it is "false", the function should not generate any plot or print anything to the command window; if it is omitted in the function call, use `output_logical = false` as default.

`towershake` does the following:

- solves the equation of motion from 0 to 20 s using ODE45 (with the default error tolerances), Euler or Heun, with zero initial displacement and velocity. If `output_logical = true`, plots the displacement of the tower top as a function of time in Figure 1
- if either Euler or Heun is the chosen method, prompts the user (with the `input` command) for a time step value `dt` and calls `my_ODE_solver` to compute the solution.

- if ODE45 is the chosen method, calculates the step size used by ODE45 as a function of the step number (try to avoid using a loop for this calculation, and think about ways to use vectors instead). If `output_logical = true`, plots the step size as a function of the step number in Figure 2 and prints the **largest step size** used by the solver in the command window in a nicely formatted way.
- assigns `shakemax` to be **last maximum** in the time interval from 0 to 20 s; in order to do that:
  1. with Euler or Heun: use the `max` command to find the maximum in the final 10% of the time window and assign `shakemax` to this value.
  2. with ODE45: use the `events` feature in `options` for `ode45` to find all the maxima of the oscillation in the time window and assign `shakemax` to the last one. See the Matlab documentation on how to use the `events` feature (note: if you are having difficulties using this feature, you can use the same procedure as with Euler or Heun (described above), but with a 5-point penalty).

Note that you will have to transform (by hand) the 2nd-order equation into a system of two 1st-order equations of the type:

$$\begin{cases} \dfrac{du^{(1)}}{dt} = f^{(1)}\left(u^{(1)}, u^{(2)}, t\right) \\ \dfrac{du^{(2)}}{dt} = f^{(2)}\left(u^{(1)}, u^{(2)}, t\right) \end{cases}$$

and you will have to write a function that defines this system (to be used with ODE45, Euler, and Heun); your function should have the form:

`function dudt = vibration(t,u)`,

where `dudt` is a ***column*** vector. This function should define all the parameters of the tower model and include comments to describe these parameters and their units. This `vibration` function needs to be a **nested** function inside the `towershake` function.

Analysis and report:

1. use `towershake` (with `output_logical = true`) to solve the equation of motion with ODE45 for a 3 Hz excitation and include the resulting displacement plot in your report, the step size plot, and the largest step size value
2. use `towershake` (with `output_logical = true`) to solve the equation of motion with Euler and Heun for a 3 Hz excitation, using the largest step size employed by ODE45 (determined in the part 1 above), and include the resulting displacement plots in your report
3. Compare the shape of the 3 plots (they should be in 3 separate figures: not all 3 of them plotted together), the 3 values of `shakemax` and discuss your results

4. write a script called `freq_sweep` to study the response of the system as a function of frequency. This script should:
   - compute the response of the system from 0.1 Hz to 4 Hz in steps of 0.02 Hz using `towershake` and ODE45 (this may take a few minutes to run; make sure to suppress all plots and prints, or you will lose 5 points).

- plot the maximum displacement of the tower top (`shakemax`)as a function of frequency in Figure 3; include this plot in your report
- this plot should exhibit a large peak at a particular frequency, a phenomenon known as resonance; your script should calculate and print the values of the resonance frequency and the resonance amplitude in the command window in a nicely formatted way; include these values in your report. Discuss the implication of an excitation whose frequency is close to the resonance frequency.
- plot the time response of the turbine at the resonance frequency using `towershake` and ODE45 and include this plot in your report. Compare this plot to the one corresponding to the 3 Hz time response.

Instructions for Computer Project 3 submission:

Include all 4 m-files and your **PDF-formatted report** in a single zipped folder named **CP3_LASTNAME_FIRSTNAME** and upload to Canvas by the due date and time above. Please use the **.zip** format only to avoid problems with opening your folder zipped in other formats.

Make sure you follow the M-file formatting instructions and the report guidelines found in the Files Section of Canvas.

Please note the following, valid for all MATLAB assignments:

1. **Refrain from using the commands `clear`, `clc`, `clf` and `close` (or any of their variations) in any of your m-files, as they interfere with the grading process.**

2. **When your code is run, it should produce only what is asked for in the assignment. Please make sure to suppress any unnecessary intermediate result in any form: no other figure, curve or value printed in the command window should appear.**

   Specifically, we should only see, in that order:

   When `towershake(3,'ODE45','true')` is run,

   - **Figure 1 with displacement plot**
   - **Figure 2 with step size plot**
   - **Maximum step size value in command window**

   When `towershake(3,'EULER','true')` is run,

   - **Figure 1 with new displacement plot (Euler plot only, ODE45 erased)**

   When `towershake(3,'HEUN','true')` is run,

   - **Figure 1 with new displacement plot (Heun's plot only, Euler erased)**

When `freq_sweep` is run,

- **Figure 3 with maximum displacement as a function of frequency**
- **Resonance frequency and resonance amplitude in command window**
- **Figure 1 with new displacement plot (ODE45 only)**
- **Figure 2 with new step size plot**
- **Maximum step size value in command window**

You will lose points if you ignore these instructions.

Please note that Canvas will only accept a file with a ZIP extension: all of your M-files and the PDF report must be in that zipped folder. If we cannot open your files because you failed to follow these instructions, you will get a zero for this assignment.

You can upload your folder as often as you like, in case you find an error in your work. Older files will be kept in Canvas, but **only the last submitted folder will be graded**. It is your responsibility for this (and all other) electronic assignment to make sure we are grading the correct folder.

Finally, please keep in mind the policies about late assignments and collaboration outlined in the syllabus, and copied here for your convenience:

- <u>Late assignments:</u> homework and projects will be accepted up to 24 hours after the deadline with a 50% penalty. This policy will be strictly enforced and no submission will be accepted after 24 hours. Hard copies of homework will be due at the end of class. ***It is your responsibility to make sure that you have successfully uploaded and submitted <u>all</u> the required files in the required format to Canvas on time***: please double-check to avoid having to re-submit your work after the due date. The only acceptable proof that you have submitted your work on time is the time stamp of your Canvas submission. ***Files will not be accepted by email even if they show a "last modified date" that is before the due date.***

- <u>Collaboration:</u> students may *discuss* their assignments with each other, but homework and projects must be *completed individually* by each student. ***You must turn in your own work***. Copying someone else's work and submitting it as your own will not be tolerated. In particular, ***for Matlab assignments***, this policy means that students can discuss aspects such as the general approach to solve a problem or the syntax of a specific command, but ***they should not look at each other's codes***. If it is suspected that this has occurred, you will be reported to the Dean of Students for an honor code violation.