ME 2016 - Computing Techniques - Fall 2018-A
Computer Project 4

Electronic submission part: due Tuesday, December 4 at 2:55 pm (5 minutes before class time)
Handwritten part: due Tuesday, December 4 at the end of class

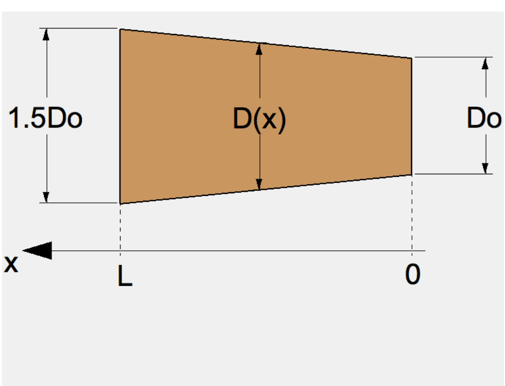
Ordinary Differential Equations: Boundary Value Problem

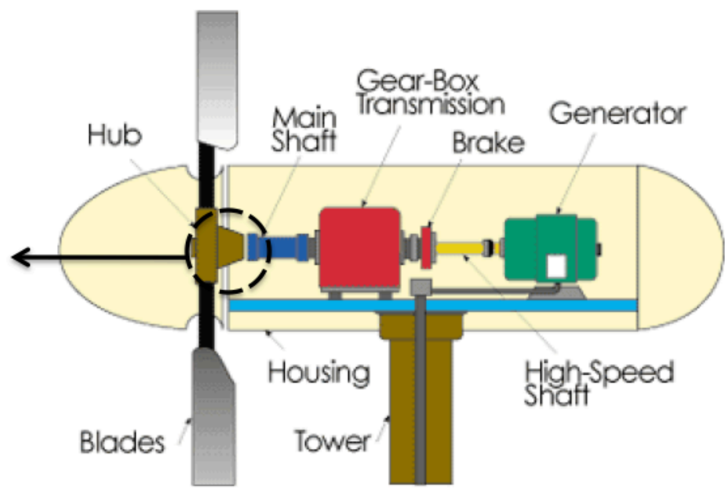
You have now moved to Denmark, a country that relies on wind power for over 40% of its electricity production, where you work for Vestas Wind Systems A/S, the largest wind turbine company in the world.

In order to save cost and to reduce the overall mass of the turbines, you want to see if the material of some of the components can be modified. In particular, you are interested to see if the steel hub and the steel shafts that transmit the mechanical power to the electrical generator can be replaced with aluminum parts. In this project, you will perform a torsional vibration analysis to determine whether or not this is feasible.

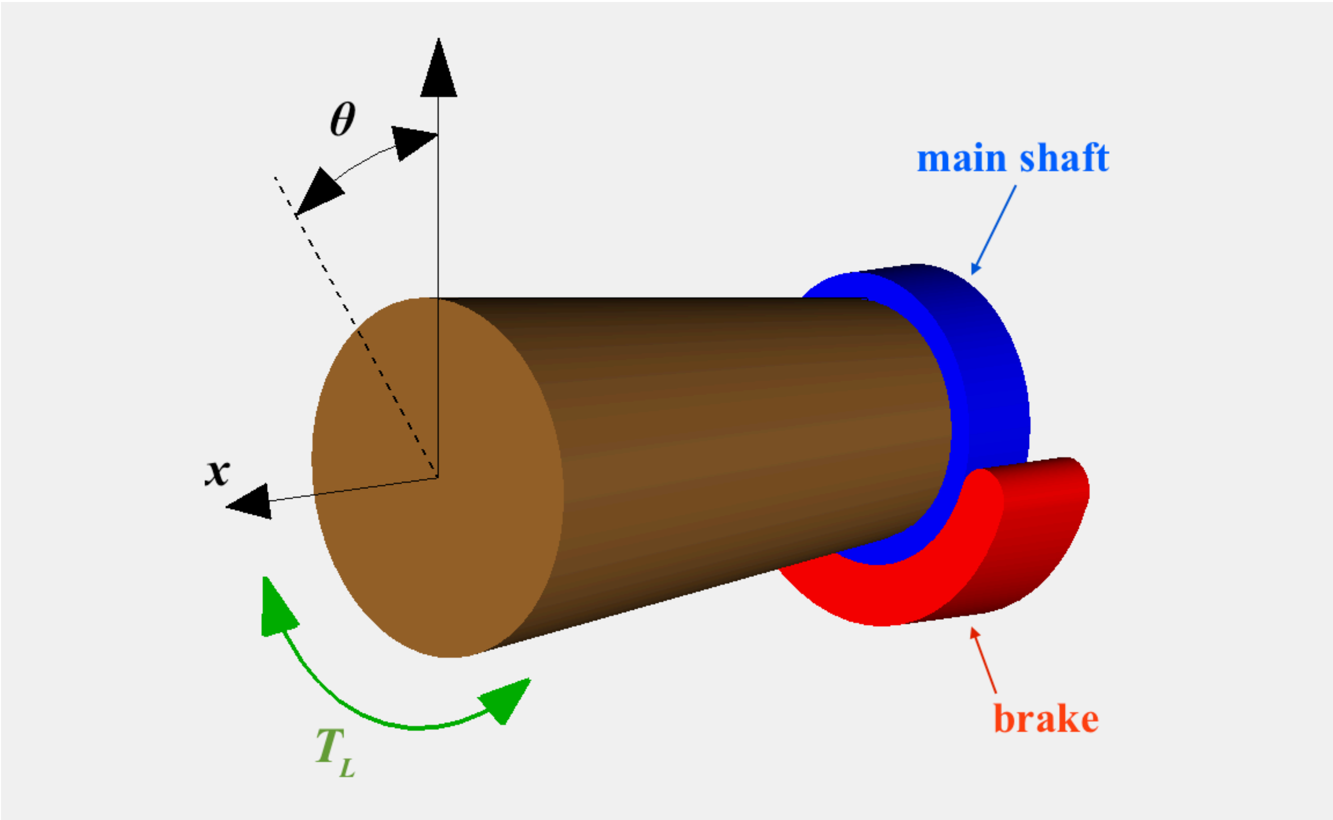
Problem description:

You decide to focus first on the conical part of the hub, shown in the drawings below. This part can be considered to be a shaft of length L with a circular cross-section and a variable diameter $D(x) = D_0\left(1+\dfrac{x}{2L}\right)$.





You want to compute the torsional vibration of this shaft, as measured by its twist angle $\Theta(x,t) = \theta(x)\sin(\omega t)$, in response to a periodic torque $T(t) = T_L \sin(\omega t)$ applied by the wind at $x = L$ (end connected to the blades), when the brake is applied (i.e., the rotor is stopped). This is illustrated in the figure below. You will compute the twist angle amplitude $\theta(x)$ as a function of the position x along the conical shaft, assuming that, in your proposed new design, a second brake (in addition to the one featured in the above drawing) is applied directly to the main shaft, and, when this brake is on, it completely fixes the main shaft.



The steady-state vibration of the conical shaft is governed by the following differential equation

$$I_p \frac{d^2\theta}{dx^2} + \frac{dI_p}{dx}\frac{d\theta}{dx} + I_p \frac{\rho\omega^2}{G}\theta = 0 \,,$$

where the twist angle amplitude $\theta(x)$ is expressed ***in radians*** and where I_p is the polar moment of inertia of the shaft cross-sectional area, given by

$$I_p(x) = \frac{\pi D_0^4}{32}\left(1 + \frac{x}{2L}\right)^4 .$$

D_0 is the diameter of the shaft at $x = 0$, G is the shear modulus of the shaft material, ρ is its density and ω is the angular frequency of the applied torque.

The boundary conditions consist of no twist at $x = 0$ (main shaft fixed) and the restoring internal torque of the shaft being equal to the applied external torque at $x = L$; these conditions are mathematically expressed as

$$\begin{cases} \theta(0) = 0 \\ GI_p(L)\dfrac{d\theta}{dx}(L) = T_L \end{cases}$$

You will use a finite-difference formulation to solve the equation for the twist angle, and you will check your results with `bvp4c`, using the following values: D_0 = 0.2 m, f = 3 Hz ($\omega = 2\pi f$), $T_L = 2 \times 10^5$ N.m, $G_{aluminum} = 24 \times 10^9$ Pa, $\rho_{aluminum}$ = 2700 kg/m^3

Hand-written part:

1) Show that the differential equation can be written as

$$\theta''(x)+\frac{2}{L}\left(1+\frac{x}{2L}\right)^{-1}\theta'(x)+\frac{\rho\omega^2}{G}\theta(x)=0$$

2) Consider the shaft discretized into a mesh of *n* segments of length $h = L/n$, with end nodes $x_0 = 0$ and $x_n = L$. Using centered-difference approximations for the first and second derivatives, express the differential equation from 1) in terms of a generic node x_i ($i = 0,1,\ldots,n$), and show that it can be written as

$$\theta(x_{i-1})\left[1-\frac{h}{L}\left(1+\frac{i}{2n}\right)^{-1}\right]+\theta(x_i)\left[A-2\right]+\theta(x_{i+1})\left[1+\frac{h}{L}\left(1+\frac{i}{2n}\right)^{-1}\right]=0$$

where $A=\dfrac{\rho\omega^2h^2}{G}$.

3) For the case $n = 4$, sketch the mesh and write the corresponding system of 3 equations for each interior node (i.e. for i = 1,2 and 3); keep n and h as generic variables in your equations (i.e. don't substitute numbers).

4) For the case $n = 4$, add an imaginary node x_5 to your sketch and use it with a centered difference approximation to express the boundary condition at $x = L$. To simplify your notation, use $B=\dfrac{512T_L}{81G\pi D_0^4}$.

5) For the case $n = 4$, write down the equation from 2) at the end node x_4 and combine that equation with the one from 4) so that it involves only the nodes x_3 and x_4.

6) Combine the system from 3), the equation from 5) and the boundary condition at $x = 0$ to show that, for $n = 4$, the equation can be expressed in matrix form $[M](\theta) = (V)$, where (θ) is a column vector of unknown values of $\theta(x_i)$ (i = 1,2,3,4) at the interior nodes and at the last node,

$$(V)=\begin{pmatrix}0\\0\\0\\-2hB\left(1+\frac{h}{L}\left(1+\frac{4}{2n}\right)^{-1}\right)\end{pmatrix}$$

and $$[M]=\begin{bmatrix} A-2 & 1+\frac{h}{L}\left(1+\frac{1}{2n}\right)^{-1} & 0 & 0 \\ 1-\frac{h}{L}\left(1+\frac{2}{2n}\right)^{-1} & A-2 & 1+\frac{h}{L}\left(1+\frac{2}{2n}\right)^{-1} & 0 \\ 0 & 1-\frac{h}{L}\left(1+\frac{3}{2n}\right)^{-1} & A-2 & 1+\frac{h}{L}\left(1+\frac{3}{2n}\right)^{-1} \\ 0 & 0 & 2 & A-2 \end{bmatrix}$$

Deriving this by hand will help you code the general matrix in Matlab for any value of n; it is also a good practice problem to prepare for the final exam.

<u>MATLAB programming</u>:

- Write a function called `torsion` with the following format:

```
function maxtwist = torsion(L)
```

This function calculates the maximum twist angle `maxtwist` as a function of the shaft length `L`. `torsion` starts by defining all the parameters and their units for the problem. It then does the following:

- Computes and plots (using circles connected by lines) the twist angle ***in degrees*** as a function of position along the shaft, using the finite-difference approach with 4 segments. Use the `subplot` command to plot this result in the top half of a figure window divided into two parts.
- Implements a loop to repeat the finite difference computation with the number of segments doubled at each iteration (i.e., 8, 16, …) until the ***relative percent error*** between the maximum angle at iteration i and the maximum angle at iteration (i+1) falls below 0.1%.
- Plots the twist angle in degrees obtained from ***the last*** FD iteration on the same plot using circles of a different color connected by lines (include a legend specifying the number of segments used).

- Computes and plots (with a continuous line) the twist angle in degrees using `bvp4c` with a relative error tolerance of 10^{-4} and an absolute error tolerance of 10^{-7}; place this plot in the bottom half of the figure window.
- Assign `maxtwist` to the maximum angle calculated by `bvp4c`

To do the finite-difference computation:

Write a ***nested*** function inside `torsion` to solve the finite-difference problem with a number of n segments. This function should be called:

```
function [nodes,theta] = finite_difference(n)
```

where `n` is the number of segments, `nodes` is the vector of x_i values and `theta` is the corresponding vector of angles. The function should define the matrix [M] and the vector (V) in the general case of n segments and solve for (θ) using the left matrix divide: M\V (note that there are better and more efficient ways to solve a system of equations in Matlab, but they are beyond the scope of this course).

Use this nested function for all of the finite difference calculations, including n = 4 (i.e., do not code the case of 4 segments separately).

To use bvp4c:

Write two ***nested*** functions inside `torsion` to define the boundary value problem and the boundary conditions; these functions should be called

`function dydx = twist_BV(x,y)`, and `function res = twist_BC(ya,yb)`

and they are to be used with `bvp4c`.

- Finally, write a short script called `CP4` to:

    1. call `torsion` and plot the twist angle as a function of position for L = 2 m (plot the 2 subplots that you get directly from `torsion`) in a figure window numbered 1; print the maximum twist angle to the command window in a nicely formatted way.

    2. find the required length of the shaft so that the maximum angle does not exceed 2 degrees (this is a root-finding problem) and plot the corresponding 2 subplots (directly from `torsion`) in a figure window numbered 2; print the result to the command window in a nicely formatted way.

There is no report for CP4.

Instructions for Computer Project 4 submission:

Hand-in hard copies of the hand-written part in class.

Include your 2 m-files in a single zipped folder named **CP4_LASTNAME_FIRSTNAME** and upload to 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.

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, when CP4.m is run, we should only see, in that order:

    - **Figure 1 containing 2 subplots**

- **A value for the maximum angle in the command window**

- **Figure 2 containing 2 subplots**
- **A value for the maximum length in the command window**

Make sure your plots are correctly formatted, with titles,…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:

- Late assignments: 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 all 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.***

- Collaboration: 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.