ENGN 2020:

Midterm #1

Brown University School of Engineering

Assigned: February 25, 2019, Due: March 1 (4pm), 2019

Instructions: You may not discuss this examination with any person or instructor, inside or outside this class. This includes all forms of communication, such as email, chat, etc.

Turn in your assignment either as a single PDF to Canvas or a single stapled document by 4pm on the due date. Your assignment should be clearly formatted, with all Figures and Tables logically numbered. Code should be pasted as an appendix when explicitly requested.

This examination is worth 50 points in total. Please turn in your well-documented code, as appropriate, for consideration in partial credit.

Problem 1

[15 points]

Consider a system of masses and springs as discussed in lecture[1], where the force on any mass j can be given as

$$F_j = -\sum_{i=1}^{N} k_{i,j} q_i$$

where q_i is the displacement of mass i from its equilibrium position. The masses are constrained to move only in a single common dimension, x. The masses (in kg) and spring constants (in N/m) for our system are given in the table below. Note that by the physics of the problem $k_{ij} = k_{ji}$.

j	m_j
1	8
2	2
3	3
4	5
5	9

$k_{i,j}$	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
$j=1$	8	3	7	1	0
$j=2$		7	5	0	1
$j=3$			9	3	1
$j=4$				5	3
$j=5$					10

(a) What are the natural frequencies (ω) of the vibration of this system? Also report the associated eigenvector with each frequency. Clearly state the units for each.

(b) Consider displacing the masses by the following amounts:

j	1	2	3	4	5
q_j	0.1	0.05	0.01	-0.3	0.3

Express the position of the system in terms of the natural movements of the system; that is, in terms of the eigenvectors. Be sure that your basis vectors have an ℓ_2 norm of 1; that is, they are normalized.

(c) Calculate and report the force on each mass as a result of the displacement vector given above.

[1] See §1.4.2 of the online notes if you need to review.

Problem 2

[15 points]

In class, we examined a very simplified version of the PageRank algorithm, which services like Google use to rank-order websites. If you read the reference given[2] it notes that there are two major shortcomings of the algorithm in its simplest form. You should familiarize yourself with these issues, and examine the improved version given in Section 3.1 of that article.

Here, we will use this algorithm to propose rankings of sports teams, focusing on College Football, where hundreds of teams play in disparate conferences across the country, but a unified ranking system is needed in order to place teams in Bowl Games at the end of the season.

We will examine the 2005 football season, and produce week-by-week rankings according to the algorithm. The game-by-game data is available from a sports statistics website[3] in spreadsheet form. You can download the data yourself; however, I have extracted the key data and put it on Canvas in "json" format, which is relatively easy to open in python. There is also an example script to help you understand how to load the data.

You should structure your approach just like the algorithm of Section 3.1, but instead of tallying links from one website to another, count each game as a link, with the link going from the losing team to the winning team.[4]

(a) To show that you can implement the modified algorithm from Section 3.1 of the reference, use it to assign importance scores to the 5-page network of Figure 2, except delete the link from page 3 to page 4. (*Hint:* Try to structure your approach so that it will be easy to adapt it to the "big data" problem we are about to do.) Report the importance scores of the five pages for m values of 0.15, 0.25, and 0.35. Normalize these importance scores by the ℓ_∞ norm of the eigenvector; that is, divide by the maximum value so that the highest-valued entry is always 1.

(b) Now perform rankings on the college football data. Start by examining only week 1, which contains 55 games. Use $m = 0.15$. When your power method converges, you should have only two distinct scores; report these values.

(c) Now extend your analysis to perform week-by-week (cumulative) rankings. *E.g.*, for week 5 your $\underline{\underline{\mathbf{A}}}$ matrix should contain all games played from weeks 1 to 5. Report the top 10 teams at the end of the season, in a table along with their importance score. For the team that finished at the top of your rankings, plot their importance score versus week of the season.

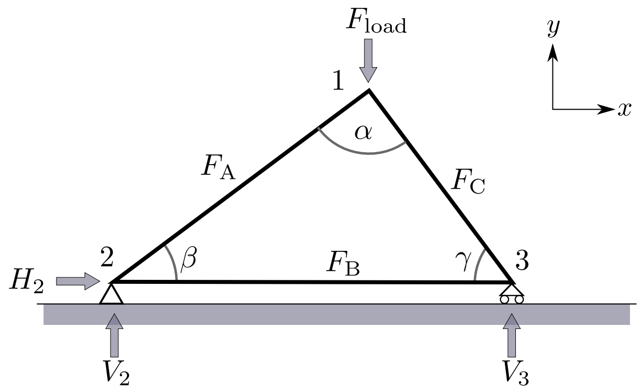
[2] Bryan, K.; Leise, T. The $25,000,000,000 Eigenvector: The Linear Algebra behind Google. *SIAM Review* **2006**; 48, 569581.

[3] `http://www.cfbstats.com/`

[4] If American football is unfamiliar to you, it is simply the team with the higher score that wins. There are no tie games in the data set provided.

Problem 3

[20 points]



The above is a truss. H_2, V_2, and V_3 are the horizontal and vertical forces that the support must exert to keep the truss stationary; node 3 is on a roller, so is only supported by a vertical force (V_3), while node 2 is rigidly attached to the support and has both horizontal and vertical forces supporting it (H_2 and V_2). A vertical load is applied to node 1.

When the system is at rest, we can analyze this system by requiring that the forces on each node sum to zero. For example, the horizontal (x) components of the forces on node 3 can be written as:

$$0 = -F_{\mathrm{B}} - F_{\mathrm{C}} \cos \gamma$$

By convention, the forces along each member (F_{A}, F_{B}, F_{C}) are taken to point into the structure; thus, in the above equation both horizontal forces bear negative signs from the perspective of node 3. H_2, V_2, and V_3 point in the directions shown on the diagram. Similarly, the vertical (y) force balance on node 2 is:

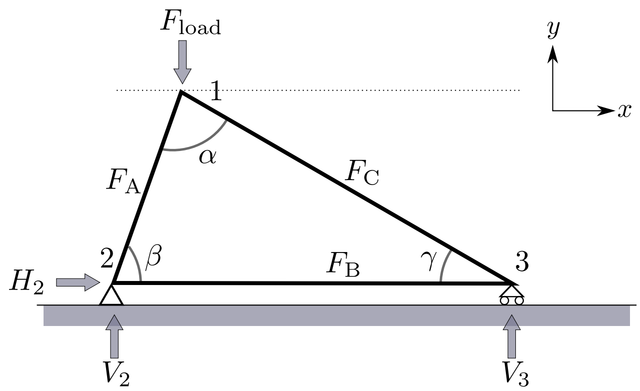
$$0 = V_2 + F_{\mathrm{A}} \sin \beta$$

(a) Write the remaining four force balance equations; there will be two force balances (x and y components) per node.

(b) Convert this into a linear system ($\underline{\underline{\mathbf{A}}}\ \underline{\mathbf{x}} = \underline{\mathbf{b}}$), where the unknown vector $\underline{\mathbf{x}}$ contains the six unknown force components ($\begin{bmatrix} F_{\mathrm{A}} & F_{\mathrm{B}} & F_{\mathrm{C}} & H_2 & V_2 & V_3 \end{bmatrix}^{\mathrm{T}}$).

(c) If the external load is 1000 N, and $\alpha = 90°$, $\beta = 30°$, and $\gamma = 60°$, numerically solve the linear system and report the six unknown force components.

(d) We would now like to understand the range of allowable angles (α, β, γ) that we can use to design the truss. Assume that nodes 2 and 3 are fixed, and the height of node 1 is fixed, but we have the freedom to choose its horizontal position, in order to vary α, β, and γ. Thus, if we specify one angle, such as γ, the other two angles are set.



The maximum weight that can be supported by the roller at node 3 is 800 N. Find the maximum angle, γ, such that this value is not exceeded; you must use a non-linear solver to get full credit for this part. Assume the load (F_{load}) is fixed at 1000 N.

Hint: Write a function that takes in a value of γ, then computes α and β and solves the linear system in order to predict V_3. Have the function return $V_3 - V_{3,\text{max}}$, such that the function value is zero when the limiting value of γ is reached. You can then use a non-linear solver, such as `scipy.optimize.newton`, with your function to search for the value of γ that solves this system.