Mitchell Truss:

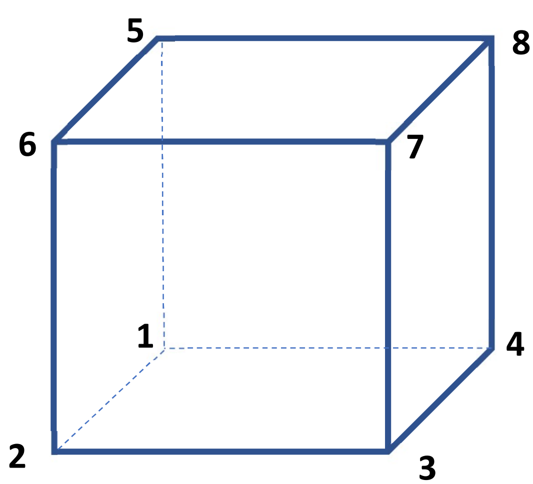
A) A structure is potentially inconsistent in two conditions, where the rank (r) of the matrix for the conditions of static equilibrium (A_{se}) is less than the number of directions (d) times the free nodes (q) or where r is less than $d * q$ however r is equal to the number of bars (b) plus the number of strings (s). The structure is not potentially inconsistent, so A_{se} is linearly independent and the system is solvable without needing to introduce soft modes.

B) A structure is underdetermined in two conditions, when r is less than $b + s$ or when $b + s$ is less than r but r is equal to $d * q$. The structure is not underdetermined, so there will be one solution for the system. In fact, this structure was neither underdetermined nor potentially inconsistent, as A_{se} is a square matrix with independent rows and columns.

Nonminimal Prism 4 Bars:

A) The structure chosen had members attached to every other node in a rectangular prism then the nodes highest on the z axis were rotated 45°. The structure is potentially inconsistent, as r is less than $d * q$ $(18 < 28 * 8)$. This means that A_{se} has linearly dependent spaces. As a result, soft modes are introduced.

B) The structure is underdetermined, as r is less than $s + b$ $(18 < 24 + 4)$. This means that A_{se} has less constraints than variables. Therefore, the system had ten degrees of freedom for motion. The results of the transegrity_statics algorithm found that some strings were not under tension, which indicates that the system is over constrained. By removing the strings on the top and bottom of the cube, there were less degrees of freedom, but the structure was still both underdetermined and potentially inconsistent (NonminimalPrism_try2.m). After removing even more strings, such that there were no longer any degrees of freedom, and the structure became potentially inconsistent but was no longer underdetermined (NonminimalPrism_try3.m). Underdetermined systems can be actively controlled, so depending on the application these degrees of freedom may or may not be desirable. All the Nonminimal prisms were potentially inconsistent



Homework #1:

Join GitHub using your email
https://github.com/
Download and configure git on your computer:
https://help.github.com/en/github/getting-started-with-github/set-up-git

Then, send email to Muhan (muz021@ucsd.edu) (note: not to Tom…) with your GitHub username and email address, so Muhan can invite you to join the MAE290a-fall2019 organization (he will respond relatively quickly). DO THIS IMMEDIATELY, as you can't really start this homework until you are added to this organization.

Once you have joined the organization, fork the repository
https://github.com/MAE290a-fall2019/tensegrity-statics
using a new repository name HW1.lastname.firstname (replace lastname and firstname with your last and first names…)

Take it for a test drive by running (in Matlab on your computer) the examples provided, including the main designs in the following paper:
http://renaissance.ucsd.edu/multiple_tether_stabilization.pdf

You will submit your homework via writing/uploading three files to this repository. You will write two new codes, MitchelTruss4.m and NonminimalPrism4.m, and a homework writeup, HW1report.pdf, and will upload these three files to this revised repository. Note: DO NOT EMAIL YOUR HOMEWORK TO EITHER ME OR MUHAN. Part of this assignment is figuring out how to use GitHub.

Muhan will have office hours this week Tu 2-3pm and Th 4-5pm, in Jacobs Hall 1603 to help you set this up if you have questions.

Also, download Bob's book, available here:
https://www.researchgate.net/publication/268640584_Tensegrity_Systems

The actual work part of homework is as follows:

Leveraging the tensegrity_statics and tensegrity_plot codes provided from my GitHub page, following the examples provided, write a new test code, MitchelTruss4.m, for a Michell Truss of order 4, given as Figure 4.6 In the pdf of Bob's book (note that the nodal points on the circle at the left are taken as fixed nodes P_i, whereas all other nodal points are taken as free nodes Q_i).

Also, write a new test code, NonminimalPrism4.m, for a non minimal tensegrity prism with 4 bars as shown in Figure 3.43 of Bob's book, but with 4 bars (all nodal points are taken as free nodes Q_i in this design).

Discuss both tensegirty structures in your HW1report, specifically including the following for each:
A) Is this structure potentially inconsistent? What does that mean?
B) is this structure undetermined? What does that mean? If it is underdetermined, is it pretensionable, or only tensionable under load? Discuss.

Also include any other salient points regarding these designs. You are encouraged to explore well beyond the specific questions asked; extra credit will be granted as appropriate...

Update to transegrity code
Hi all -
I just updated all of the codes at
 https://github.com/MAE290a-fall2019/tensegrity-statics
They are easier to use (less buggy, and prettier plots) now. Please re-download all of these codes.
Thanks,
Tom
PS - These codes seem to work well on 2019 versions of Matlab, but not on 2016 versions of Matlab. Please update your Matlab version if you haven't recently; I believe you can get it for free here:
 https://matlab.ucsd.edu/student.html

HW1 Hints
Hi all -
Curiously, a couple of 290a students have expressed to me significant angst about the current HW. I think maybe they have overestimated what is actually left to be done in this homework. It actually isn't that hard, as I've already done the bulk of the coding work for you, and provided you with several working examples. You have already been reading my paper. You just need to study how these example codes work (focus on TBar.m and TBar3.m, which are the simplest 2D and 3D examples provided), then reference two tensegrity designs described in Bob's book and, following these my examples, define the free and fixed points, Q and P, and connectivity C of the b bars and s strings in the corresponding tensegrity structures. Then, define some test loads U for each structure, run my tensegrity_statics and tensegrity_plot codes (feel free to fiddle with both if necessary, though it likely won't be...), and analyze the results.
Since this homework is actually pretty easy, I have left it kind of open-ended. Once you get the above working, please feel free to get creative. Bob's book has a number of other tensegrity designs that you can study/extend using these codes, and you have an opportunity to get some significant extra credit for so doing. I aspire through this experience to give you a taste of what it is like doing research - grabbing a hold of an idea, digging into the related scientific literature, putting the appropriate numerical codes in place (implementing various fundamental 290a-related linear algebra concepts), then exploring and seeing where the research leads.
most of all, have fun with it!! :)
- Tom