Tensegrity | cuboctahedron

July 28, 2017

These *tensegrity*[1] structures are beautiful but can be cumbersome to make by hand. Perhaps that is part of the beauty in a way, but to me the real beauty is the often surprisingly elegant mathematics behind the configurations. Here we give a very quick derivation of the already small Python program towards the cuboctahedron structure like seen in fig.1[2]. For some notational clarifications, have a look

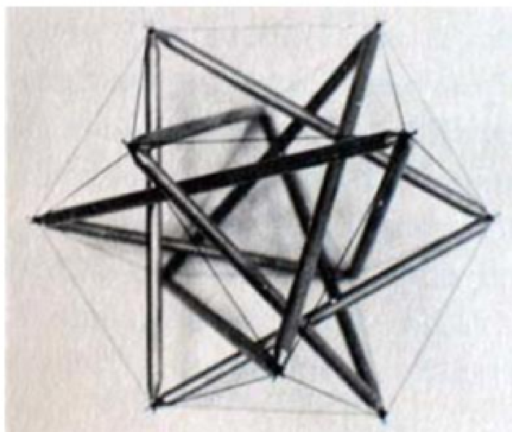

Figure 1: *"4 triangles nucleated in a 24 tensile vector network, from Structural Morphology Of Tensegrity Systems by Motro, based on a model by Pugh."*

at fig.2 to see what is meant with radius and triangle leg length and "cord length".

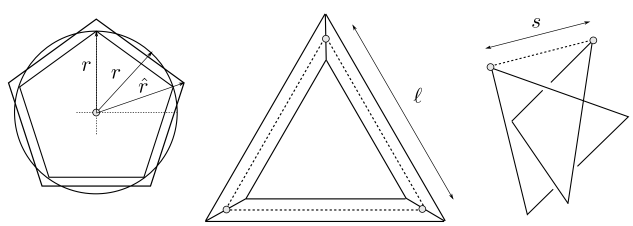

Figure 2: *What is meant by the radius(r) and the triangle leg length(ℓ). Cord length(s) is the length between two points, from different triangles. We will see that the cuboctahedron its edges essentially are the "cords" we are looking for.*

Given the cord length, compute the triangle leg length

Lets have a look. The goal is to find ℓ, the length of a triangle leg, given s, the (main) cord length. In this way you can for example 3D print your triangle, with

[1] https://www.google.nl/search?q=tensegrity
[2] http://tensegritywiki.com/Cuboctahedron?showComments=1

inhumane precision, where-after hopelessly fooling around with the cord lengths is over, since that was the input! We will see that you can say for example, I want s to be 6cm, then the program delivers you the design for the triangles which work with $s = 6$cm(easy to measure), this happens to be $\ell = 10.3923$(not so easy to measure yourself).

To give a quick and simple derivation have a look at fig.3. By Pythagoras $h = \sqrt{2}s$, by Euler $t = \frac{1}{2}\sqrt{3}s$ and again Pythagoras $w = \sqrt{t^2 - (\frac{1}{2}h)^2} = \frac{1}{2}s$. Given our coordinate system, $p_1 := [-\frac{1}{2}s, \frac{1}{2}s, \frac{1}{2}\sqrt{2}s]$, $p_2 := [s, 0, \sqrt{2}s]$. Then compute the length of $\ell = \|p_1 - p_2\|_2 = \sqrt{\frac{9}{4}s^2 + \frac{1}{4}s^2 + \frac{2}{4}s^2} = \sqrt{3}s$. Thus $\ell = \sqrt{3}s$, quite cool!

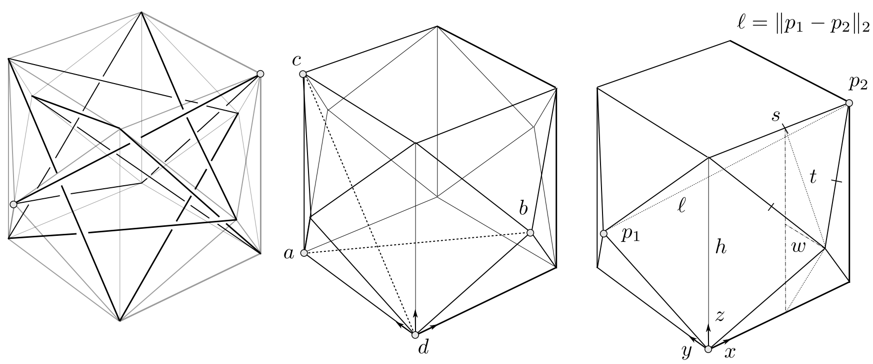

Figure 3: *The polyhedral(cuboctahedron) structure this tensegrity structure is based on.*

Given the triangle leg length, compute the radius

You can also do it without any cord/thread/rope and make the legs touch. This contact allows for the structure to stay upright. What we need to answer is, given two overlapping lines within our cuboctahedron, which will(should) touch in the printed structure, what is the minimal distance between them? Then the effective radius of the triangle leg will be that distance divided by two.

See the middle picture in fig. 3 with $a := [0, s, \sqrt{2}s]$, $b := [\frac{1}{2}s, -\frac{1}{2}s, \frac{1}{2}\sqrt{2}s]$, $c := [0, s, \sqrt{2}s]$, $d := [0, 0, 0]$. Describe the lines by $a + (b - a)\theta_1$ and $c + (c - d)\theta_2$. Now we want to find the θ's such that the distance between those two lines is minimized. Intuitively this sounds like a convex problem, and it is. The problem can be cast as a least squares problem through

$$\underset{\theta_1,\theta_2}{\text{minimize}} \quad \frac{1}{2}\|a(1-\theta_1) + b\theta_1 - c(1-\theta_2) - d\theta_2\|_2^2.$$

Rewrite this with $B = c - a \in \mathbb{R}^3$, $\theta = [\theta_1,\ \theta_2]^\top \in \mathbb{R}^2$, $A = [b - a,\ c - d] \in \mathbb{R}^{3\times 2}$:

$$\underset{\theta}{\text{minimize}} \quad \frac{1}{2}\|A\theta - B\|_2^2$$

which has the standard minimizer:

$$\hat{\theta} = (A^\top A)^{-1} A^\top B. \tag{1}$$

Plugging this back into the original objective and dividing by two gives the effective length we seek

$$r = \frac{1}{2}\|A\hat{\theta} - B\|_2 \tag{2}$$

Again, the relations are no black magic. Note that r relies on s, and $\ell = \sqrt{3}s$ thus the user input length is easily related to the radius.

Great, we have the radius. For printing and aesthetic reasons the leg is shaped as a pentagon, not a smooth circle! The question is, how the approximate a circle. Should we use the inner or outer circle of the pentagon, can we do better? Yes. The idea is to minimize the absolute difference between the area of the circle and that of the pentagon. In the words of fig.2, given r, find $\hat{r} = \frac{r}{c}$. This c is given by $|\sqrt{5\cos(\pi/5)\sin(\pi/5)1/\pi}|$ which really follows from the respective areas and setting $\frac{\partial}{\partial c}(5\cos(\pi/5)\sin(\pi/5)r^2 - \pi r^2 c^2)^2 = 0$.

Assembly

Some advise on the assembly. Be patient and have a good look at fig.1!

For the *floating* version: Since s is set, use that! Start by setting up the main structure, like on the left of fig.3. Of course, it will fall down, but you need it. This can be done for example with some pieces of tape to keep a triangle together. Then you can connect all the *squares*(like the dashed one in middle), just the squares(6), with length s. You will see that structure come to life!

For the *contact* version it is a lot quicker but also here it could be easier to use some tape to put the final triangle together where-after you can firmly attach the edges together with for example some thread. The holes are actually very convenient for assembly, therefore the standard in the script.