4450 HW 5

toyonaga@g.harvard.edu

December 16, 2021

Contents

1 FDM

The force density q associated with a connection in a mesh whose nodes have coordinates $\vec{x}$ and $\vec{x}'$ respectively is given by:

$$q \equiv \frac{|F|}{|\vec{x}' - \vec{x}|} \tag{1}$$

For reference, the FDM method consists of resolving the following relation between loads p, force densities q_i, and displacements $\vec{x_i} - \vec{x}$ at every node in a mesh where i indexes nodes which are connected to the node under consideration at location $\vec{x}$.

$$\vec{p} + \sum_i q_i(\vec{x_i} - \vec{x}) = 0 \tag{2}$$

1.1

Given the coordinates provided we can simply plug into 2 and solve. We treat the direction of gravity as parallel to $-\vec{z}$ and treat each dimension independently:

$$0 + 1(0 - x) + 1(5 - x) + 1(0 - x) + 1(7 - x) = 0 \rightarrow x = 3 \tag{3a}$$

$$0 + 1(0 - y) + 1(0 - y) + 1(7 - y) + 1(5 - y) = 0 \rightarrow y = 3 \tag{3b}$$

$$-5 + 1(0 - z) + 1(3 - z) + 1(3 - z) + 1(0 - x) = 0 \rightarrow z = \frac{1}{4} \tag{3c}$$

1.2

Given the definition of the force density 1 we can compute the force in each member:

$$\begin{aligned} F_{tot} &= \sum_i F_i \\ &= \sum_i q_i |\vec{x}' - \vec{x}| \\ &= \frac{17}{4} + \frac{\sqrt{321}}{4} + \frac{\sqrt{329}}{4} + \frac{\sqrt{521}}{4} \end{aligned} \tag{4}$$

1.3

We repeat the exercise in 1.1 with $q_2 = q_4 = 2$.

$$0 + 1(0 - x) + 2(5 - x) + 1(0 - x) + 2(7 - x) = 0 \rightarrow x = 4 \tag{5a}$$

$$0 + 1(0 - y) + 2(0 - y) + 1(7 - y) + 2(5 - y) = 0 \rightarrow y = \frac{17}{6} \tag{5b}$$

$$-5 + 1(0 - z) + 2(3 - z) + 1(3 - z) + 2(0 - x) = 0 \rightarrow z = \frac{2}{3} \tag{5c}$$

We see that the equilibrium position of the node is "pulled" towards p_2 and p_4. This makes sense since the increased force-density corresponds to increasing the stiffness/reducing the rest length of the corresponding connection.

1.4 Matrix Representation

1.4.1 Branch-Node Matrix

We define a branch-node matrix C following [1]. Since the mesh has 7 nodes and 10 edges, C has shape $(10, 7)$[1].

[1] Note that the reference follows a highly unusual notation in which the first index of a matrix C_ij corresponds to columns. We instead adopt the usual convention of indexing rows with i and columns with j.

$$
\mathbf{C} = \begin{pmatrix}
-1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 & 0 & 0 \\
0 & 0 & -1 & 1 & 0 & 0 & 0 \\
-1 & 0 & 1 & 0 & 0 & 0 & 0 \\
-1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 1 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & -1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 1 \\
0 & -1 & 1 & 0 & 0 & 0 & 0
\end{pmatrix} \tag{6}
$$

1.4.2 Force Densities

The force density matrix Q is a diagonal matrix of shape $(e, e) = (10, 10)$ containing the force densities.

$$
\mathbf{Q} = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 7 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10
\end{pmatrix} \tag{7}
$$

1.4.3 Node Positions

In the node positions can be represented as three vectors of length $n = 7$ associated with each dimension. (We could of course combine these into a matrix with shape $(7, 3)$):

$$\mathbf{x} = (0, 1, 2, 3, 5.2, 3, 3) \tag{8a}$$

$$\mathbf{y} = (0, -2, -1, 0, -1, -2, -3) \tag{8b}$$

$$\mathbf{z} = (0, 0, 0, 0, 0, 0, 0) \tag{8c}$$

1.4.4 Loads

Similar to the node positions, we can define the loads as a set of 3 vectors of lenth $n = 7$ corresponding to forces applied to the nodes in each direction. We set the load to 0 for fixed nodes.

$$\mathbf{p_x} = (0, 0, 0, 0, 0, 0, 0) \tag{9a}$$

$$\mathbf{p_y} = (0, 0, 0, 0, 0, 0, 0) \tag{9b}$$

$$\mathbf{p_z} = (0, 1, 1, 0, 0, 1, 0) \tag{9c}$$

1.4.5 Fixed/Free Nodes

The fixed nodes are associated with $i = 0, 3, 4, 6$. The Free nodes are associated with $i = 1, 2, 5$.

1.4.6 Branch-Node sub-matrices (problem 4.f, 4.g)

One can permute the columns of C such that the first n' columns are associated with free nodes. We have:

$$\mathbf{C}' = \begin{pmatrix} 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{10}$$

We can identify the first 3 columns with $\mathbf{C_N}$ and the final 4 columns with $\mathbf{C_F}$

1.4.7 Load Matrix

Permuting the rows of $\mathbf{p}_i$ we can write down the corresponding loads:

$$\mathbf{p_x}' = (0, 0, 0) \tag{11a}$$
$$\mathbf{p_y}' = (0, 0, 0) \tag{11b}$$
$$\mathbf{p_z}' = (1, 1, 1) \tag{11c}$$

1.4.8 Matrix FDM

We can write down a linear system for the unknown coordinates:

$$\mathbf{C_N}^T \mathbf{Q} \mathbf{C} \mathbf{x} + \mathbf{p} = 0 \tag{12}$$

Equivalently using einstein notation (with repeated indices summed):

$$C_{N,i\nu} Q_{ij} C_{jk} x_k + p_\nu = 0 \tag{13}$$

We see that a given row i of this equation is simply the FDM equilibrium equation 2 for the ith free node in the mesh.

We can of course rearrange this equation to highlight the unknown quantities:

$$\mathbf{C_N}^T \mathbf{Q} \mathbf{C_N} \mathbf{x_N} + \mathbf{C_N}^T \mathbf{Q} \mathbf{C_F} \mathbf{x_F} + \mathbf{p} = 0 \tag{14}$$

	q	grid divisions	metric
0	1.00	2.0	66.93
1	1.00	3.0	86.41
2	1.00	4.0	104.74
3	1.00	5.0	122.23
4	1.00	6.0	139.27
5	0.75	2.0	56.00
6	0.75	3.0	73.04
7	0.75	4.0	89.53
8	0.75	5.0	105.22
9	0.75	6.0	120.63
10	0.50	2.0	46.74
11	0.50	3.0	61.14
12	0.50	4.0	76.34
13	0.50	5.0	90.41
14	0.50	6.0	104.56

1.5 Computational FDM

We implement the fdm algorithm as a `flask` application that can be called in a grasshopper script using `hops`. Code can be found at link. Example derived forms are shown in the following section.

1.6 Additional Form finding

We parameterize our system by modulating q, effectively the mesh spring constant and the mesh pitch (number of divisions given fixed edge length in original configuration). Example structures are shown in 1

1.7 Force Analysis

We compute the edge forces as in **??** for the structures found in 1.6 and report the forces in **??**.

We observe that structures that have a shallow shape are less efficient than structures with a higher aspect ration (tall/skinny). Additionally, decreasing the stiffness of the springs used to parameterize the force density system leads to more efficient structures, since this creates these taller/skinnier structures.

References

[1] Sigrid Adriaenssens et al. *Shell structures for architecture : form finding and optimization*. London ; New York: Routledge/ Taylor & Francis Group, 2014.

q = 1

q = .75

q = .5

Figure 1: Sample structures found by varying q and the number of divisions

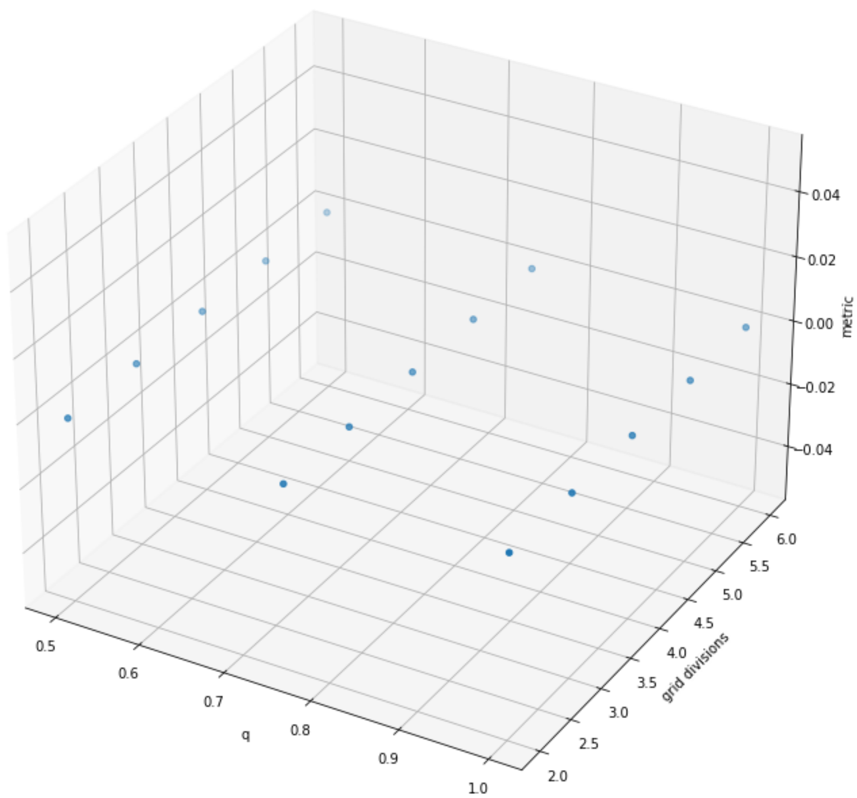

Figure 2: Performance Metric of form-finding structures