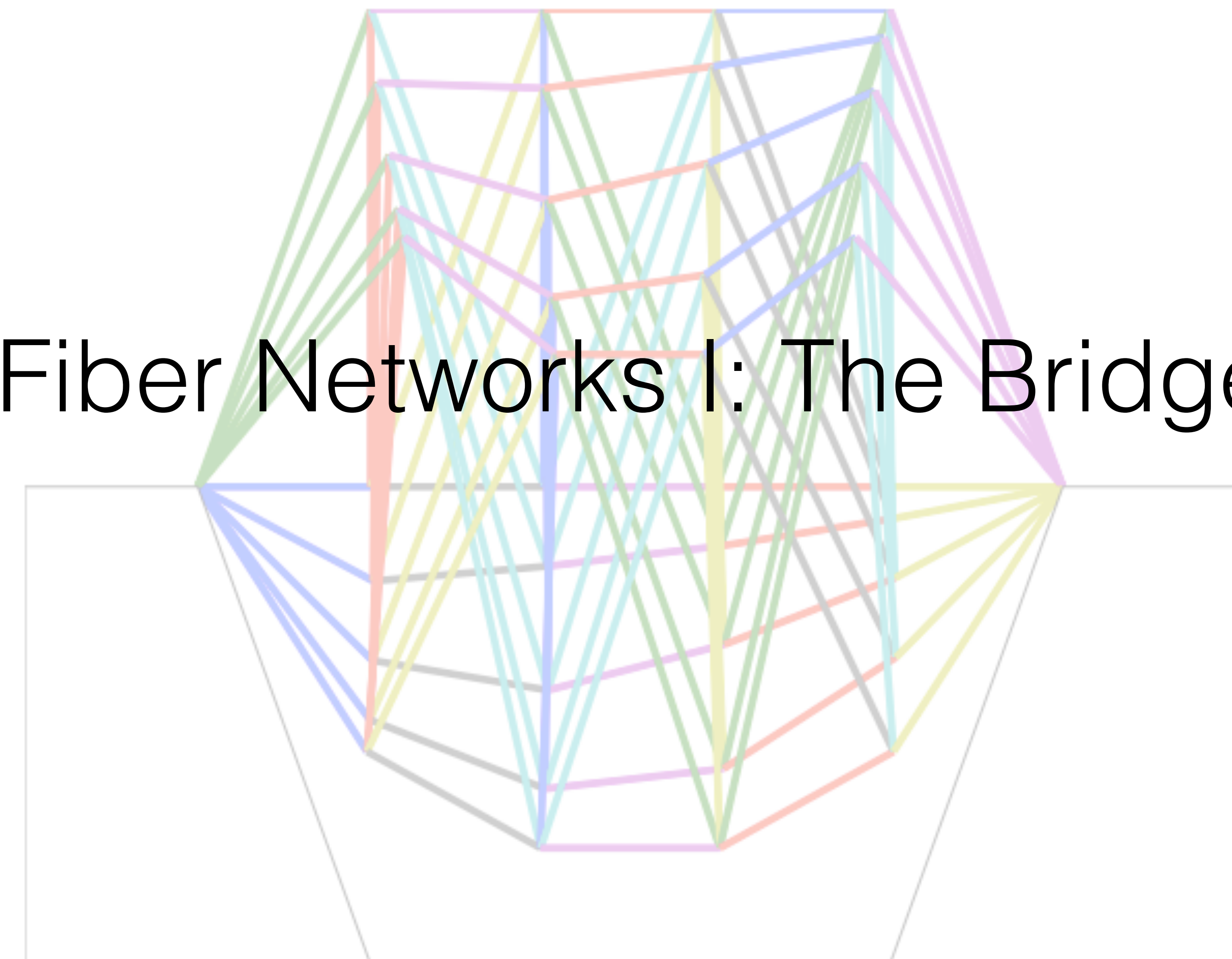
Fiber Networks I: The Bridge



One function: bridge.m

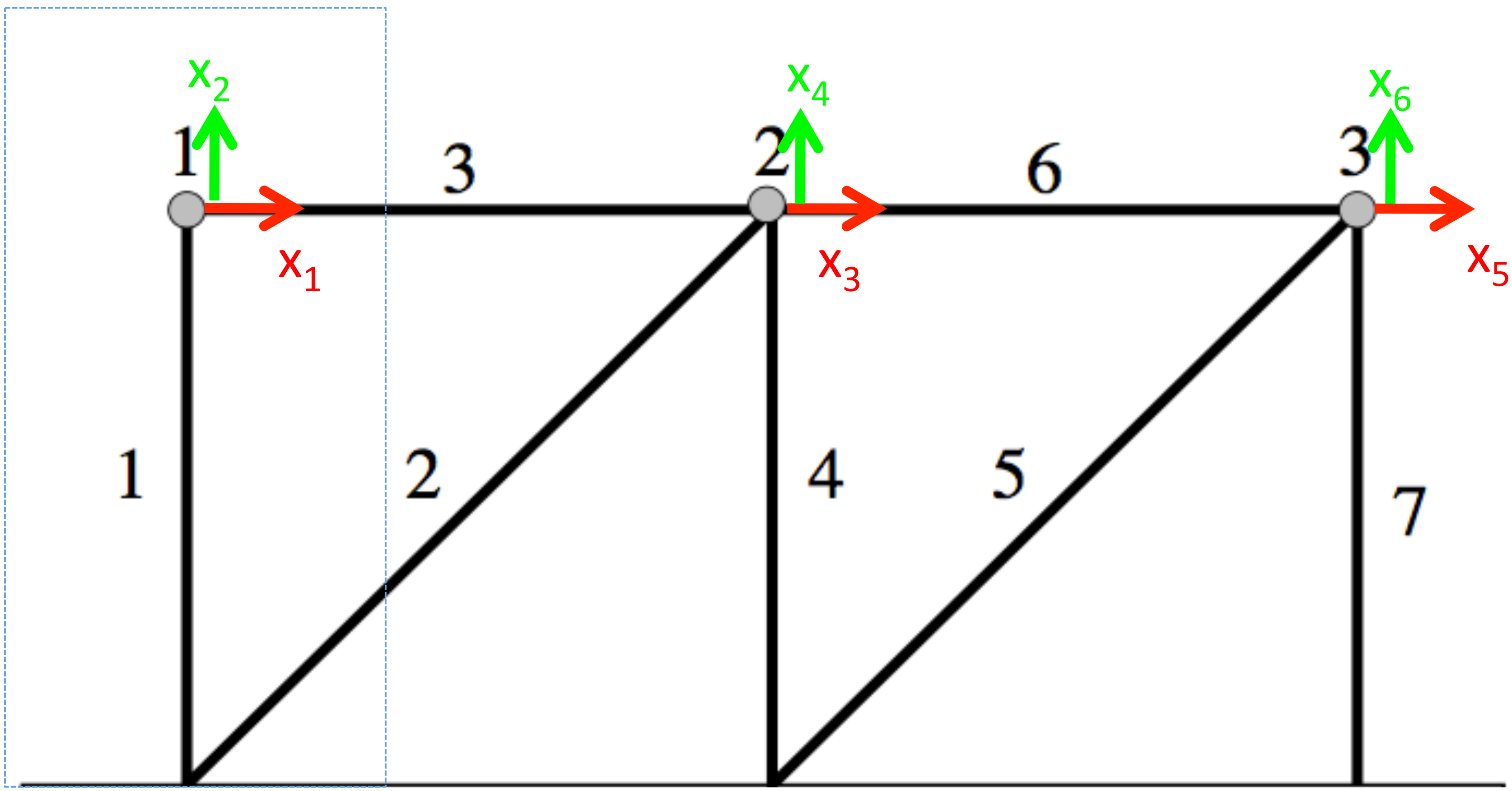
Needs to accomplish 3 major goals:

One function: bridge(Ea,W,nos)

Needs to accomplish 3 major goals:

- Build adjacency matrix (address bridge in undeformed state)… automate except for first and last stages!
- Find coordinates of undeformed bridge
- Find coordinates of deformed bridge

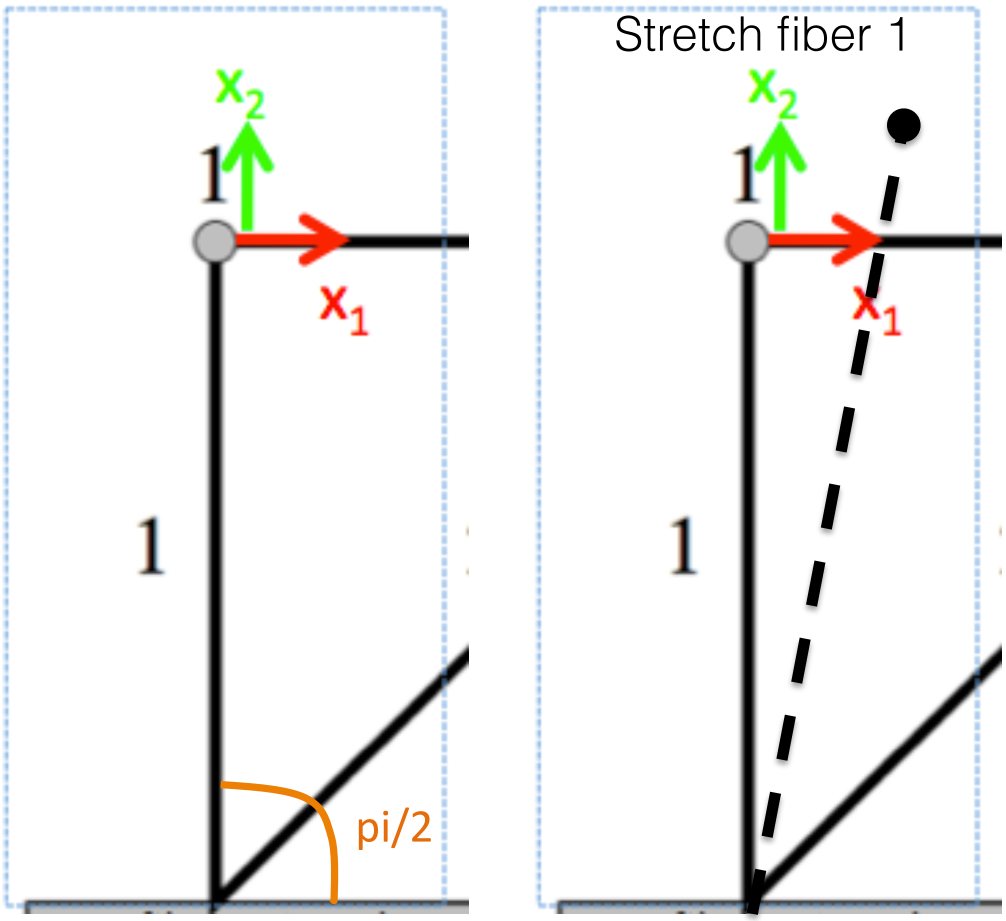
The Adjacency Matrix: some review



7 fibers; 3 nodes. When we consider the elongation of a *fiber*, we evaluate what's going on at the *nodes* at each of its endpoints. Note that some of its endpoints may not be associated with nodes if they're connected to the edge.

When you get a new structure, always start by drawing it *by hand* and labeling your degrees of freedom at each node.

Calculating Elongation



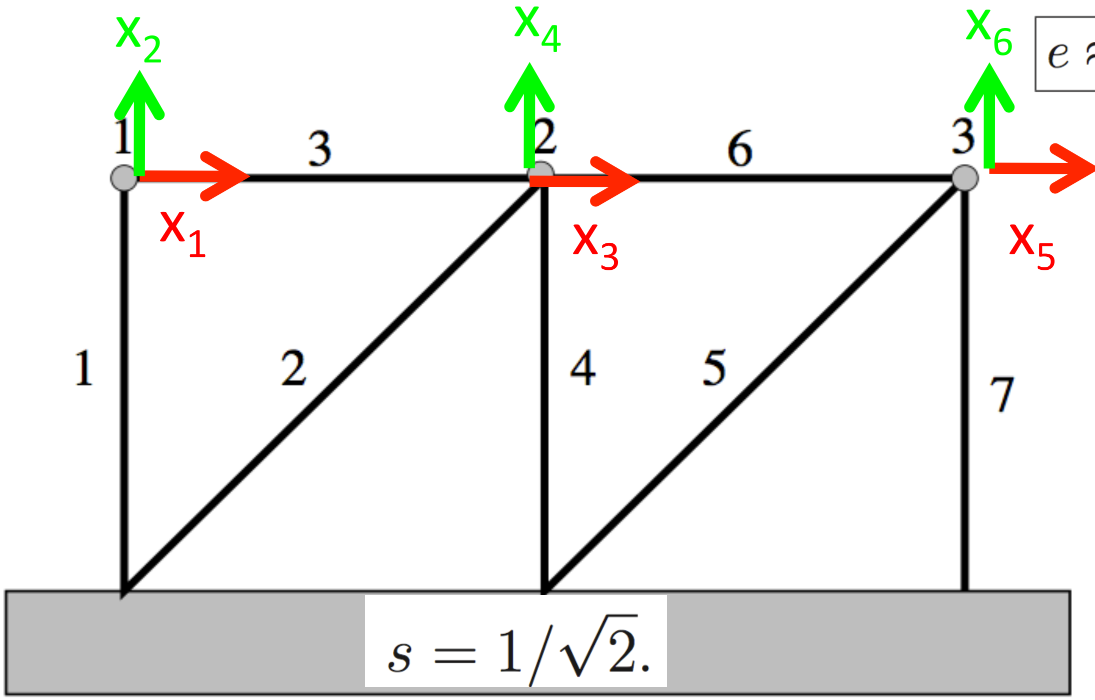
Fiber 1 is only associated with one node, so we don't have to consider subtraction in this simple case.

We have:
e1=(x1)cos(90) + (x2)sin(90)
=(x2)sin(90).

Don't get confused about the signs and subtracting. Think of it more as an absolute difference in lengths than a literal difference. If you are actually subtracting when there are multiple nodes, just be *consistent* in subtracting head-tail or tail-head, it doesn't matter which.

$$\boxed{e \approx (x_3 - x_1)\cos\theta + (x_4 - x_2)\sin\theta.}$$

$$\boxed{e \approx (x_3 - x_1)\cos\theta + (x_4 - x_2)\sin\theta.}$$



$$e_1 = x_2$$
$$e_2 = (x_3 + x_4)/\sqrt{2}$$
$$e_3 = x_3 - x_1$$
$$e_4 = x_4$$
$$e_5 = (x_5 + x_6)/\sqrt{2}$$
$$e_6 = x_5 - x_3$$
$$e_7 = x_6.$$

$$\begin{array}{c} \\ e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \end{array}
\begin{array}{c} \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array} \\
\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & s & s & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & s & s \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Each column in A corresponds to a degree of freedom. You put in that entry the *coefficient* of that degree of freedom in the list of elongations you made above.

Each row in the adjacency matrix corresponds to a *fiber* (the elongation of that fiber, $e_1 \ldots e_N$)

Why?

Because we want to be able to write e=Ax…

…and write our elongations as a system of equations in terms of $x_1 \ldots x_n$

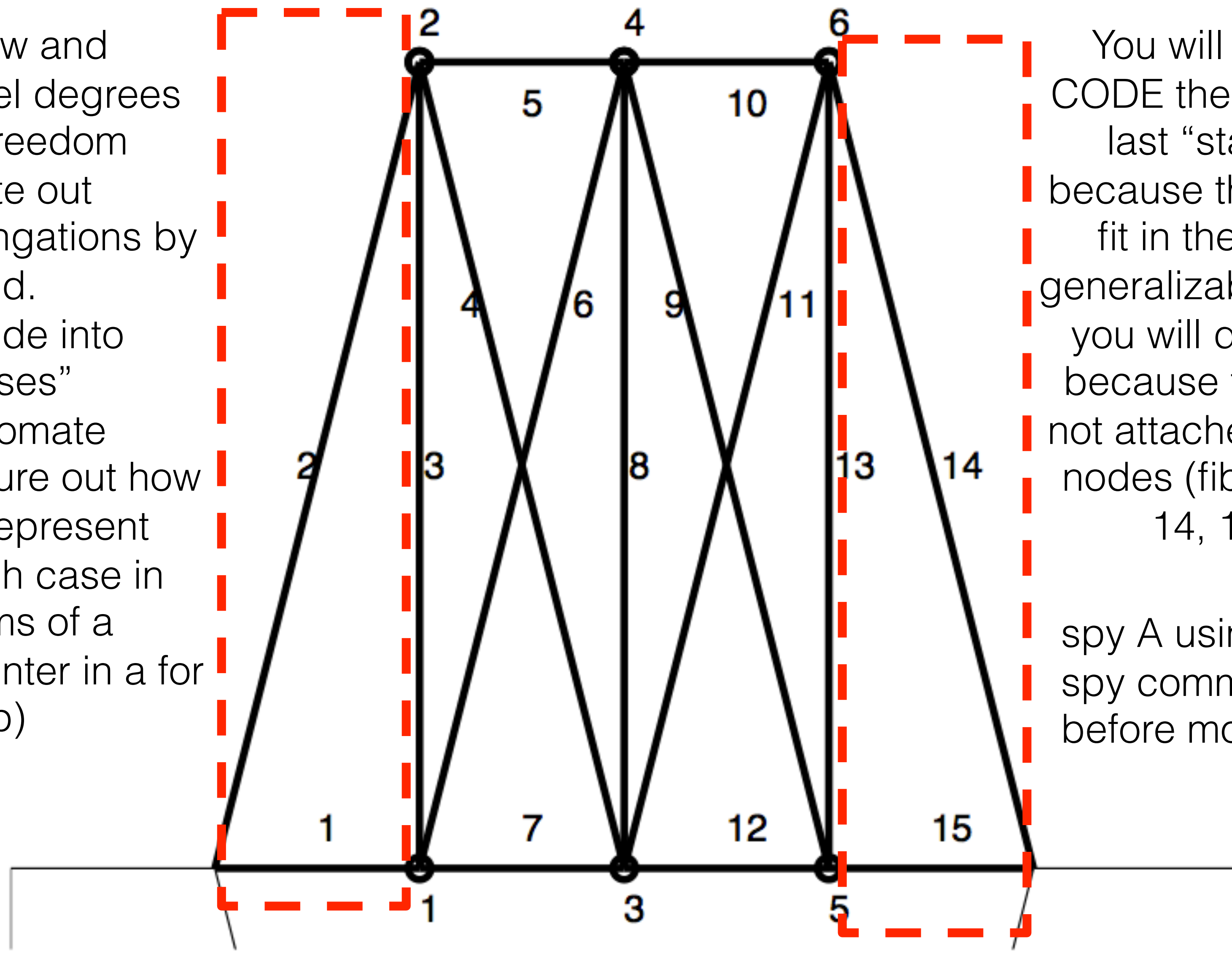
$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \\ e_8 \end{bmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & s & s & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & s & s \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}$$

$$e_1 = 0x_1+1x_2+0x_3+0x_4+0x_5+0x_6$$
$$= x_2,$$

$$e_2 = 0x_1+1x_2+sx_3+sx_4+0x_5+0x_6$$
$$= sx_3+sx_4$$

Our project: adjacency matrix

1. Draw and label degrees of freedom
2. Write out elongations by hand.
3. Divide into "cases"
4. Automate (figure out how to represent each case in terms of a counter in a for loop)

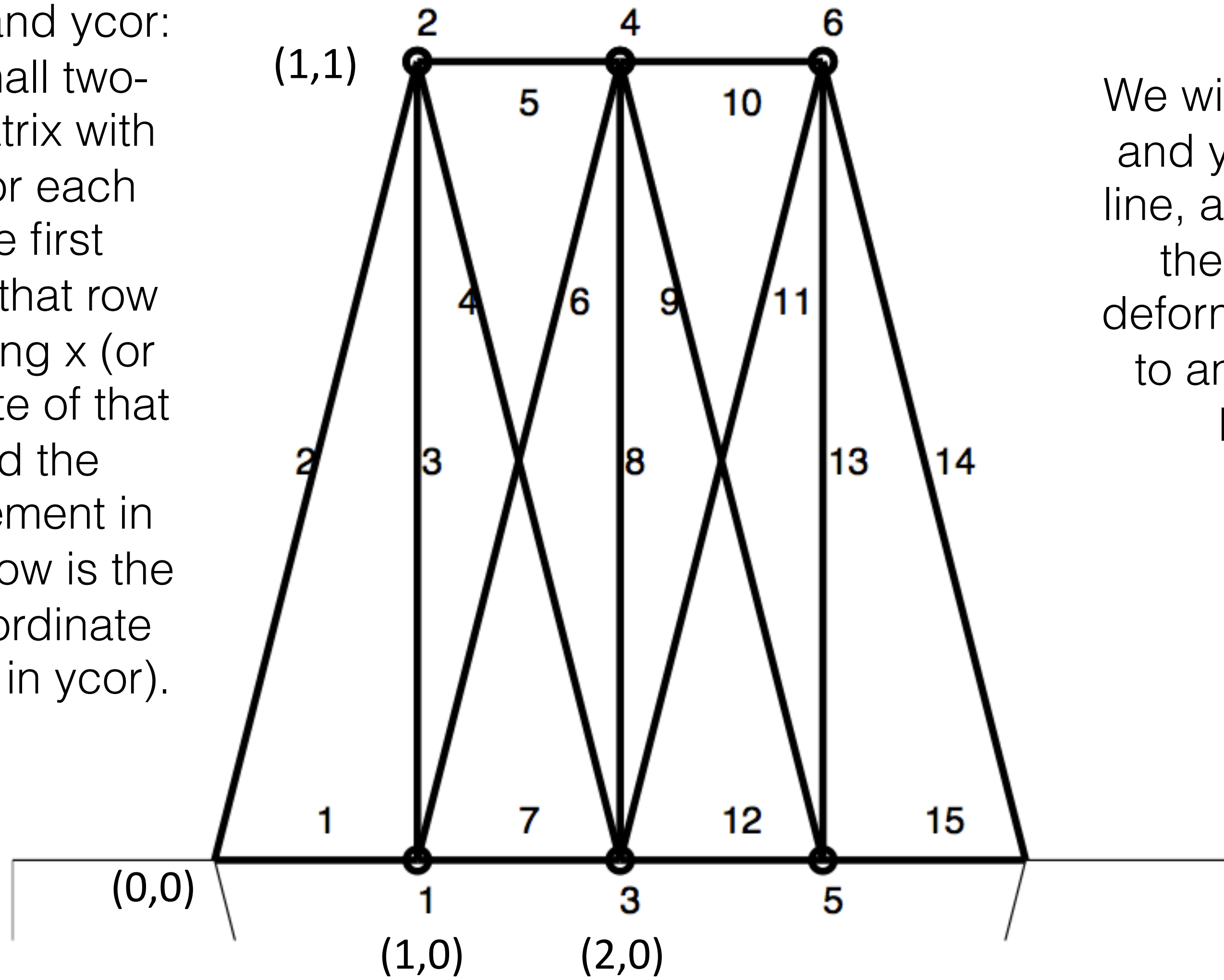


You will HARD CODE the first and last "stages" because they don't fit in the other generalizable cases you will discover because they are not attached to two nodes (fibers 1,2, 14, 15).

spy A using the spy command before moving on!

Undeformed coordinates

Build xcor and ycor: each is small two-column matrix with one row for each fiber. The first element in that row is the starting x (or y) coordinate of that fiber, and the second element in that fiber's row is the ending coordinate (x in xcor, y in ycor).



We will plot xcor and ycor using line, and update them upon deformation due to an applied load.

Updating Coordinates upon Deformation

- You have xcor and ycor, the undeformed coordinate matrices. It’s a problem of addition: add something to those coordinates under different loads and plot new coordinates (xcorold +change=xcornew) to see deformation. Not conceptually difficult.

How to implement?

- First you need to solve for the displacement itself. Remember

$$\underbrace{A^T K A}_{S} x = f.$$

- A is adjacency matrix,
- K is a diagonal matrix of fiber stiffnesses. Remember the stiffness of a fiber = Ea/L. We need a vector of lengths of each of the fibers, then we can compute a vector *k*, and make this vector the diagonal of a matrix K.
- We then use MATLAB's built-in solver x=S\f to find displacements (x). Remember we have 4 different *f* vectors depending on where the cars are (do this four times and superimpose plots).

Updating xcor and ycor

Once we have these displacements, we can add them to the original coordinates to find the deformed coordinates, then plot those.

Again, we hard-code the beginning and ending stages, and automate everything in the middle (because the middle we can again divide into “cases.”

What form is the otuput of x=S\f?

Updating xcor and ycor

Once we have these displacements, we can add them to the original coordinates to find the deformed coordinates, then plot those.

Again, we hard-code the beginning and ending stages, and automate everything in the middle (because the middle we can again divide into “cases.”

What form is the otuput of x=S\f?

x = 1 x dof vector: $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ \dots \end{bmatrix}$ $\leftarrow$ movement (displacement) at degree of freedom x1

→we need to update xcor and ycor based on movement in the x direction (odd elements of x) and movement in the y-direction (even elements of x).