PROFESSOR STRANG:
OK, so this is the second, last lecture on trusses. Then we've got a holiday on Monday. And then after that we'll be into Chapter 3. I thought I'd write down just in case it's any use to you, the four problems that I intend to include with the next homework. That won't be due for quite a while, a week from Monday. So these will be the problems on trusses that come from particular trusses drawn in the book. And then there'll be some problems from the new material, that we do next week. So trusses and really, there's two main jobs for today. One is to identify this matrix A, the strain-displacement matrix or the stretching matrix. How far do the bars stretch? Everybody remembers A is going to come in this step. If we have displacements, then, of the nodes like, this would be like a u_1, this would be a u_1^H and a u_1^V, this would be a u_2^H and a u_2^V, so there are four movements of the ends of the truss and of one particular bar, and then we'll stretch that bar. And the question, is how much?

So that will be one row of A. So if we follow one bar, you remember in the matrix A, there's going to be a row for every bar. So a row for each, row of A. For each bar. And if we track down one of those rows, we'll have the idea. And then of course at the end we'd maybe be constructing A without, sort of a free-free A. And then at the end, any fixed displacements that will knock out columns of A. So that's one job. And then to see, so the A is going to be a little more messy. It's because we're in two dimensions. So compared to the network problems, and and the line of springs, now we have more happening. We've got more columns because every node has now two unknowns. A horizontal and vertical. So A is kind of bigger. And therefore A transpose C A, you might think, it's going to be hard to see what's going on. But you'll see the right way to look at A transpose C A is a bar at a time. That's the nice fact about A transpose C A, I might focus on that first.

And then comes the fun part. I'll draw some more trusses, that may or may not have mechanisms. They may or may not be stable. And we can try to identify the mechanisms. Actually, as before. We'll do it by engineering instinct rather than by solving. I mean, in principle, we could always use elimination. Or ask MATLAB or any other system to do it, and look for the solutions to Au=0. And decide are the columns of A independent. In that case the

truss is stable. This matrix is invertible, we know all the good possibilities. And then there's the more interesting possibility, of having some solutions to that. In which case that matrix will be singular. There'll be some modes in our big system that will cause it to fail. But it's kind of fun to find those. OK, while I've written A transpose C A, may I remind you about a good way to do that multiplication. OK, so imagine I'm just putting a number. Here's going to be the matrix A. So the matrix A will have a bunch of rows, row one, row two, so on. These rows will correspond to bar one, bar two, and bar three. OK.

OK, then we have C, so that's a square matrix. That each bar has a spring constant, so c_1, c_2, c_3, and then we have A transpose. And those rows, or columns of A transpose. So that's the sort of picture of A transpose C A, for a three-bar, three bars only. But the point is made right here. There's a row of A for every bar. Right? Because our matrix A is m by n. If there are m bars, a row for every bar, and it tells us how far that bar is stretched. And we'll figure out what its entries are. That's our main job. I'm just looking ahead. Suppose we've got that row. And that row, and that row. So a row for every bar. Now, here I've taken three bars. Now, how do I multiply those matrices? Well, I can do it different ways. But here's a cool way to do it. Just-- The way I want to point out is column times row. If you multiply matrices you're allowed to-- The effective c_1, c_2, c_3, is going to be very simple. So I'm really paying attention here to A transpose A. If I want to multiply A transpose A, I can do row times column as usual and get one number. Or I can do column times row and get a whole little matrix. And that's the bar one matrix. It's the element matrix, and that's how finite elements will be assembled, and that's why I should keep mentioning this point.

So the way to do that column times row thing, and then of course that c_1 just multiplies that row, that'll be c_1. Row one, transpose, that's the column, times row one. That's what's coming from bar one. That column multiplies the c_1 and that row. You see how nice, that's the element matrix associated with the first bar. And then there'll be a second column times the c_2, times the second row. So plus c_2, row two transpose, row two. That's a matrix again. Plus c_3, row three transpose, row three. I focus on that because you don't think of this as a way to multiply matrices, but it's really a nice thing to notice. And it's better to notice it now when we have three bars or something, than in a big finite element code. Yeah. So this is, just if I complete it, complete this thought, this I would call a one bar matrix. That's the matrix A transpose A if there's only one bar. Actually, one of the problems at the end of this section is find the element matrix for one bar. And I guess it's about what we're going to get to when we do that one bar. Do you remember what it was in the, just to connect this thought, what was

the little matrix in the case of networks? So in the case of networks, there was just one unknown for each, not two. So for networks, just, I'm just going to put down the, and you'll recognize it immediately. The little element matrix was the c for that, and there was a [1, -1; -1, 1] what? Do you remember that guy that was the [-1, 1] from a row? [-1, 1] from a column?

So this was exactly c times the [-1, 1] from the column times the [-1, 1] from the row. That's where this simple little matrix came from. And you remember that the, so what's involved in creating this big A transpose C A is just create all these little pieces, which are like this, but they're going to be a little bigger. In fact, in a minute I'm going to ask you what size they'll be. Well, they're really big matrices. There are a whole lot of zeroes there that I didn't even put. Zeroes are there for rows and columns that aren't touching this particular edge. And again, this matrix. There'll be all kinds of zeroes in A. Because a typical row of A, bar one, is going to have non-zeroes only for the, yeah what's the size? How many non-zeroes in a typical row of A? Getting the count right first is like half the battle. How many non-zeroes in a typical row of A? This was the network case where we had a couple of nodes. And they were connected. And we had an unknown at each end. So two unknowns were involved. The little matrix was two by two. It properly has lots of zeroes for all the other nodes that are not involved. And then that matrix kind of gets, assembled is the word I think usually used. All these little guys get assembled, you know, pasted, stamped, I hear as the verb sometimes now. Take these little matrices for this little element. And stamp them into the big A transpose C A. This is the c_1, so this gives the c_1's in the matrix. And the c_2's and the c_3's. Alright.

Now, just before we-- I'm like doing this preliminary, before I write down anything, the exact row. What's the size of, for trusses, how many non-zeroes in a row of A? So that's my question. How many non-zeroes in a typical row, like for that bar, non-zeroes in a row of A? So A is the matrix that tells us how much, that row of A is the row that tells us how much this bar stretched when this moved along by u_{H1}, and up by you u_{V1}, and this moved along by say, u_{H2}, and up by u_{V2}. Well, I've written all those in. So that you can tell me this number. How many? What's your guess? When I tell you, you'll say of course. How many u's are involved in the stretching of that bar? Four. Four. Exactly. Instead of one at each end, we have two at each end. So the answer is four. How many, the answer is four. And now the only remaining question is, what are they? What are those four numbers? The four non-zeroes in the row? So let me just answer that. They are, so here is that row. So we the two non-zeroes associated with it. Well, the way I've numbered these nodes one and two-- Since I've numbered them one and two, the non-zeroes are going to come right at the start. And then a whole lot, then this is

all going to be zero after that. Because those will be nodes three, four, or five, whatever, that don't involve bar one. So bar one just connects node one to node two. Now, what do you think? Well, let me put in the key quantity here.

This bar is at an angle. It's at an angle theta. So there's a theta. Angle theta. OK. And so that angle is going to enter these things. In fact, here's what you get. You get, I think if I put the one up there and the two down there, let's see. What am I thinking now? I'm saying if u 1 H is positive, that's going to stretch the bar. That's a positive stretching. So I'm expecting a positive u 1 H to give me, I'm expecting that sort of to come in with a plus sign. Now suppose the bar is horizontal. Suppose the bar is horizontal, then how much does the u 1 H stretch it? It stretches it by the whole u 1 H, right? If the bar was horizontal, so theta equals zero. I'm just doing these, we got to sort out this theta angle stuff. So here's my thing. If the bar happens to be horizontal, then that stretching by u H 1, will completely stretch the bar. If the bar happened to be-- Yeah yeah. And of course, this way. So that for a horizontal bar, I'll just be back to this step. I'll have a one and a minus one. u-- Oh yeah, remind me about that. Why doesn't u vertical, for a horizontal bar like this, why does this one not stretch the bar? You remember that from last time, that was the little bit of trig that we did when we forced ourselves to stay linear. So we dropped the second order correction, that would come from going this way. Right?

I mean you must have noticed, like walking? Suppose you want to walk from here to the end of the bar, OK? Well, if somebody moves the end of the bar forward, you have to take those extra steps. The bar really stretches. But, if somebody moves the bar this way, then the extra bit of length is much less. In fact, it's zero to first order. This is like taking shortcuts when you walk across the courtyard. So when the angle's theta, I'm only expecting a one and a minus one. On the horizontal. And zeroes on the vertical. OK, now I'm ready to write it in. I think when the angle's theta, when the angle's theta, any theta, that was when the angle was zero, I think we get a cos theta. Doesn't your instinct say that this is on the u_1, u horizontal 1. And then the u vertical 1, tell me what these should be. And then we'll make, what do you suppose is the entry, second non-zero, the one that corresponds to a vertical movement. Here it would be, for a horizontal bar when theta is zero, I'm going to see a zero there.

But if the bar is at an angle like this, what am I going to see? Everybody's going to get it right? What do I put in there? Sine theta. What else could it be? Right, OK, sin(theta). And now what about the next guy, the other end of the bar? u H 2 and u V 2, those are the other two non-

zeroes. What your guess for u 2 H, u H 2? If I move this forward, what's the change in length of the bar? What would your guess be that goes into there? Say it again? -cos(theta). -cos(theta), right, yeah. The movement of the other end, like if I move this guy a little bit to this side. That will shorten the bar. Forget about that one. If I move this over, the bar becomes shorter. And the cosine tells me the key number there, how much it becomes shorter. If the bar was horizontal, the cos(theta) was one, it counts a hundred percent. If the bar is vertical, and I move it horizontally, it comes zero percent. Because the linearity says there was no first order change. And now tell me the final non-zero entry. And I see I didn't leave much room for all the zeroes.

OK, what's the u 2 V entry? -sin(theta), of course. And then come all the zeroes, for whatever other joints are not involved with bar one. So let me, maybe to make this picture best, I should move that over to where it belongs. Now, if I add up along a bar, add up the four numbers there, what do I get? Zero. You expected that, right? In fact, if I add just that and that I get zero. If I add that and that I get zero. Just the way I got zero here. In the incidence matrices. The column of all ones is certainly going to solve Au=0. Unless the supports remove those, of course. If the supports don't allow all ones because some have to stay at zero, then I could have a stable truss. OK, that's a typical bar. A typical row. That's a typical row. OK, and now maybe while I'm on the same subject, what is the size, what does this thing look like now? This is in A. This is in the matrix A, and now I want to ask you before I even come back to all this stuff, what about in A transpose A? In A transpose A, A transpose C A, it the whole deal. The element for-- The little matrix, the element matrix, can I call it that? Or the one-bar matrix, call it the one-bar matrix. Will be, is what?

So I want this, it would be typical-- c_1, row one transpose, row one, that's the typical guy. And how many non-zeroes in that? Multiplying a row. Sorry, multiplying a column that has four non-zeroes times a row that has four non-zeroes times a number, which is just fine. How many non-zeroes are going to sit in this element matrix, this one-bar matrix? 16. 16 non-zeroes. And they're going to be, I have cosine, sine, minus cosine, minus sine, multiplying cosine, sine, minus cosine, minus sine. And all multiplied by c_1. So that's the matrix. And you see what it looks like. c squared, cs, so on. 16 guys. So we have four squared as our element matrix, where here we had two squared. And in finite elements, when you get to elasticity, and you've got triangles, you've got triangular elements, then there are three nodes involved. So you're up to higher numbers. But this gives you the idea. And remember that this four by four, the way I've done it, the way I've numbered it, one, two, happens to sit up in the upper left corner.

Of A transpose C A. But can you sort of imagine how the code would be written? The code would be written, take each bar, and what do I have to know about the bar?

Just imagine a code that would do trusses. Actually, the final problem that I'm not assigning in this section says what would the code be like? Can we just have a think about what the code would look like if we were to write it. What would the input have to be? For each bar, what input do I need? For this bar, I need to know-- And for the whole truss. What do I have to tell, what's the information that I need for the whole truss? I have to know the positions of all of the joints, right? So I'd have to know the coordinates of that, (x, y), the coordinates of this one, (x_1, y_1) for joint one, (x_2, y_2). So I'd have to have a little list of what would that be? m by two? Oh, no. N. N by 2. I have N, what do I have now? Think of what information do I have to report about this truss? I guess I have N, capital N, joints. And I need two coordinates, x, y for each position. So that's N by 2, this.

OK, and then for every bar, what do I need to tell it? What do I need to put in the code for a typical bar? I certainly have to put in the c for that bar. And what else do I need to know? I need to know which joints it's connected. Right? I have to tell the system that this bar is between two and one, one and two. I have to tell it which pair. So I guess I have a list of m bars, and for each bar I must tell the system the two node numbers, and the c, the stiffness. The constant for Hooke's Law. Right, do you see this picture? Just sort of visualizing, creating a code here. And then the code would do all this, oh, have I given enough information to find theta? Or do I have to import theta also? No. I told you the positions, so it'll figure out cos(theta) theta and sin(theta). It actually won't figure out theta, that's always a dumb thing to do, find the actual angle. cos(theta) and sin(theta) is the quantities we want. So given that position, (x, y), and this position (x_2, y_2), it would know cos(theta) and sin(theta). And having drawn his picture allows me to make once more the key point about small displacements. What's the angle of the bar after it's moved? After it's displaced? It was theta before it was displaced, and the angle after is theta. To first order, the angle doesn't change. Because these are little tiny movements of the ends. I've drawn them much bigger than they should be drawn. They're little, tiny movements of the ends so that the angle is not significantly changed. Otherwise we're into geometric nonlinearity and that stuff, that makes the problem much, much harder.

OK, are you seeing sort of the picture? I guess what I haven't completely-- I've really depended more on your intuition than on a calculation to say that these are the four non-

zeroes. What did I ask you to do? I asked you to check that that was right in the extreme cases, like if theta is zero, the bar is horizontal, then we just have a 1, 0, -1, 0; vertical isn't happening. If the bar is vertical so that the angle is 90 degrees then we would have a 0, 1, 0, -1; everything's vertical. And the book draws a little picture, and computes delta l from these four small movements. And takes the leading term and sure enough it produces that row of the matrix. Gosh, I talk real fast. But do you think you could now create the stiffness? If you had a real truss, you could create the matrix A for it? C is simple, it's given to you. You could create A transpose C A? You might just want to write the command as A' * C * A or something. And let MATLAB do the thinking. But I wanted to just see what these, how this four by four piece appears in this product, from each bar. The 16 non-zeroes will appear in different positions and you told the code what those positions are. You had to give the code a local-to-global picture. This is the local picture. Watch one bar. Then it has to fit in this big n by n matrix, and that means you have to know what joints was that bar connecting. So which positions do these 16 non-zeroes assemble into? That's some time devoted to a job that I actually don't plan to require you to do. Creating this truss problem.

What I think is kind of more fun and that's, these homework problems would deal with it, is part two of the lecture. Going back to mechanisms, and now thinking about more complicated trusses. We now in principle could find the solutions to Au=0 because we now have constructed A, and we could get MATLAB to do the work or Python or whoever. But can I go to part two now and draw a truss and ask you about the mechanisms? Let's see. I guess somewhere in the problem set, but not one of the assigned ones is-- Start with those six bars. And six joints, so these are six joints. And OK, as it stands, how many-- That's a good question. As it stands, what's the shape of the matrix A? How many rows has it got? So as it stands, so I'll call it A_0, for no supports have been added. A_0, just the full matrix. Is what shape? 6 by 12. Good. 6 by 12. Is 6 by 12. OK, of course it's not stable. We know that. We haven't supported anything. So in a typical case, how many solutions to A, so I'm going to ask you how many solutions to Au=0? And what's your guess? Six. Got six equations, we've got 12 u's, 12-6, so this is going to be 12-6. And of course six solutions to that equation. It's what I would expect. There could be, it could be possible that the six equations are not independent. If they really dropped to five then this would bump up to seven, I don't think it's going to happen.

Now, can you describe those six solutions, not with numbers, just with, tell me. I hope you can, because I can't right now. OK, three of them we know. So with no supports at all, what are

three rigid motions? And what are they? The whole truss could move to the right, the whole hexagon. It could all move up, it could all rotate about one point. All three of those would be movements, displacements that don't stretch anything. OK, three rigid motions. Across, up, and rotate. OK, and I can get rid of those by supporting some nodes. But let me see, I don't know what's going to happen. When I describe this topic as the fun one in 18.085, it's more fun for you than for me. Because I draw something like that and I start worrying can I think of three-- How many mechanisms to look for? Three. That's a big number. I bet you I can find one but you guys have got to, alright, tell me some mechanisms. Let me try to draw them. What would be one mechanism? Collapses, yeah, somehow. How shall I make it collapse? Can squeeze in, yeah, maybe that's the first one. This guy comes in, this guy, what does that do? I've got 15 minutes here, I could pull that board down and draw another one. What is this one here? Let's see, if that comes in, do these guys have to go up a bit? Yeah, because that angle's not 90 degrees. So we've got a first order change in this. So this comes in a little, this goes up a little, this guy maybe stays straight. Would you go for this, I mean, please say yes?

Something happens there, right? These things go in and those go out. Could you create the-- I mean if this was a equal-side, regular hexagon, you could put in all the numbers for all 12, I would be looking for 12 numbers. Six joints and they each have two u's, so that wouldn't be so simple but you could do it. So that would be one. Looking for number two. What would be another one? So sort of squeezing in like whatever. What do you think? Any others? Maybe that's possible. Maybe I just look at it, you think that would work? We could hope those were independent, but I wouldn't put my life on it. I have this squeeze in, this squeeze in and this squeeze in, I would worry a little bit. I can see another one.

AUDIENCE: Half, for--

PROFESSOR STRANG: Fold it in half.

AUDIENCE: Along, I guess that sort of gets into 3-D, but.

PROFESSOR STRANG: Yeah, we've got to stay in the plane, right. I have instead of what? OK.

AUDIENCE: [INAUDIBLE]

PROFESSOR Squeeze that out. Yeah. Good. Number two. And then I can see, here's one, here's an easy

STRANG: one to think of. Leave these three alone and just rotate these guys. Bring this down, right? Just bring these three vertically down, or something. You see why it's sort of, did you like that one alright? It seems simple to me, looking at it, just leave these guys-- Rotation, let this turn down. This turn down, let's say, and this go down. Maybe these would all drop by the same amount. Maybe. So anyway, whatever. Let's put some supports on them. And get these numbers down. So let's support, as usual, the bottom guy. OK, so different problem now. I won't call that A_0, I'll call it A. I'll ask myself, is it stable or unstable? The matrix is now six by what? Eight, because I've taken away, I have four reaction forces, two at each support, horizontal and vertical. I've got four free nodes. And six by eight. Let me put in-- So six by eight, what am I expecting now? Any rigid motions? No, no rigid motions now. How many solutions am I expecting? Two, I think. Probably two. How many mechanisms? Well, no rigid motions. So probably two mechanisms. Now, can we find two mechanisms? Alright, this is like more reasonable. We can see whether whether we get two mechanisms and whether they're really different.

OK, what are the mechanisms now? These guys are fixed. So forget my little sketch here, and think again. What do you see? Alright, let's have one mechanism. What would one mechanism be now? There have to be two. What do you think? Sit on it. Alright, bring these guys down, and then these guys will go out, is that it? OK, so a number, mechanism number one, sit on truss. OK. Alright. Now, I don't know what number two is, that's why I'm taking time.

AUDIENCE: [INAUDIBLE]

PROFESSOR STRANG: Yeah. Or could we do this, could we bring these guys in and let's go up? It's the same thing. OK, so squeeze truss. Squeeze sides. Is that the same thing that I had, number one the same as two. Oh jeez, OK. Is that what-- Everybody is agreeing with this? OK. So I didn't get, my number two was no good. OK. What's a better number two? Hold an edge? Like that one. I'm just doing what you say, I'm not. AUDIENCE: [INAUDIBLE] Just this guy, rotates like so. And this guy will rotate, and this guy. That looks pretty good to me. Good, is that correct? Say that one again? So the first one was when these two came down and these went out. Right, OK. And now your suggestion is, you picked on this guy and held it fixed. And then this one came down a little bit. It'll, of course, how will it move? It will move perpendicular, right? Small movement. The bar is not going to change length, that's the whole point, right? The bar is not changing length. So the movement must be, it must be a simple rotation. Around here. OK. Right, and of course again you might say well, the bar really did change length because that's

not quite the same as that. But then again that's my second order basis. So that one came down, and what did this one do? Came down the same. OK, and this one it also moves. What is that? OK, so what am I going to call this one? Fix one. Yeah. Fix one node. And that makes sense. Fix one joint, yeah. And then, and rotate the rest. I think that would be possible. Yep.

OK, a small prize for anybody who, maybe handwritten, a picture of two really nice mechanisms. Somehow this one seems a little un-symmetric in a problem that's so symmetric, so I would guess that somewhere along the line we could find a kind of more some symmetric one. But I don't see what it is right now. Can the whole thing rotate a little? Could that rotate, could the whole thing rotate? Yeah, maybe it could. This guy would go up, maybe that's possible. That's somehow got everybody into the action. So I'll put "or rotate". OK, so you see what questions you get into. May I just draw a different truss? So those homework questions are other trusses. Here's one that I drew in the book itself. Yeah, may I draw this, I called it a treehouse. OK, so I have, so here's one that's actually in the book. And it's got a couple of bars going up, and one over. So that's the start. Then it's got a diagonal and that one. And then here comes the treehouse. OK, right. Well, just to get, let's again get the count right. So what's the matrix A for this treehouse? A is how many by how many? How many bars are you seeing here? One, two, three, four, five, six, seven, eight, eight bars. And how many unknown displacements? Ten.

We got five joints that are not supported, and each one has two unknowns. So A is eight by ten. So I expect two mechanisms. OK, so again I'm looking for two mechanisms. OK, what's one?

AUDIENCE: [INAUDIBLE]

PROFESSOR STRANG: It's what? This guy just falls, right. This looks unfortunately very much like the treehouses that I built for my kids. Well, so linear algebra sentenced them to fall, right? OK, that's one. I probably propped it up with one more bar, but of course that wouldn't be enough, because it's got two mechanisms, so if I make it nine by ten I haven't saved the kids. OK, with eight by ten, what's the other mechanisms? The whole thing could turn the-- Nothing preventing turning here. They can't move but they could turn. So the whole thing could go over, right, the whole thing could just-- That would be a horizontal movement of all five nodes. The horizontal of all five nodes. And again, slightly downwards, but that's a second order effect. OK, so that's the second truss. OK. So this is really like practice for discrete problems, for the problems of plane elasticity. And the point is that there are two unknowns for each point. If we have differential

equations. So the differential equations of plane elasticity are not really simple. They're not really simple. And 3-D elasticity even more. Because the points are physical points and they can move three ways, and it gets quite interesting. And those are the major problems of computational mechanics. OK, let's say, holiday time, and I'll see you next Wednesday for Chapter 3. Which moves to partial differential equations.