

MIT OpenCourseWare
<http://ocw.mit.edu>

18.085 Computational Science and Engineering I, Fall 2008

Please use the following citation format:

Gilbert Strang, *18.085 Computational Science and Engineering I, Fall 2008*. (Massachusetts Institute of Technology: MIT OpenCourseWare). <http://ocw.mit.edu> (accessed MM DD, YYYY). License: Creative Commons Attribution-Noncommercial-Share Alike.

Note: Please use the actual date you accessed this material in your citation.

For more information about citing these materials or our Terms of Use, visit:
<http://ocw.mit.edu/terms>

MIT OpenCourseWare
<http://ocw.mit.edu>

18.085 Computational Science and Engineering I, Fall 2008
Transcript – Lecture 3

The following content is provided under a creative commons license. Your support will help MIT OpenCourseWare continue to offer high quality education resources for free. To make a donation or to view additional materials from hundreds of MIT courses visit MIT OpenCourseWare at ocw.mit.edu.

PROFESSOR STRANG: Starting today on section 1.3, actually we'll finish that today. And that's about the big problem in, or the most common problem in scientific computing, solving a linear system. $Au=b$. How would you do that? So we will certainly do it for some examples. Like our free-fixed or our fixed-fixed matrices. But I thought I'd just put some comments. Often I make comments that go beyond what specific things we will do in detail. Certainly the workhorse of, if I'm in MATLAB notation, the workhorse would be backslash. That command is the quick, natural way to get the answer u . And I want to say more about backslash. I won't be able to say all the things it does. There's a lot built into that command. And the MATLAB helpdesk would give the details of all the different things that backslash actually does. It looks first, can it divide the problem into blocks, into smaller problems. It looks to see if A is symmetric. If it's symmetric, because of course, that would cut the work in half because you know half the matrix from the other half in the symmetric case. It goes hopefully along and eventually gives you the answer. Ok, so I'm going to say more about that. Because that's by elimination. The method there is elimination.

I just thought I would mention that if I had a large sparse matrix, so I was in sparse math MATLAB and used backslash, it would try to reorder the rows. Or it will try to reorder the rows in an optimal way. Now, what would be an optimal way? Just so you have an idea of what. So elimination takes this matrix, right? It's got all its rows. And we'll do examples. So the idea of elimination is that it produces a zero. So let's suppose it accepts the first row as the pivot row. And this 1, 1 entry would be the pivot. And then how does it use the pivot? It subtracts multiples of that pivot row from the other rows in order to get zeroes in the first column. So then it has a new matrix that's like it's one size smaller. And then it just continues all the way until it gets to a triangular matrix with all the pivots sitting there on the diagonal. So that's the idea of elimination. And we'll do small examples.

So why would you want to reorder the rows? Two reason, two good reasons for reordering the rows. One reason which MATLAB will always unless it has some good reason not to, is if that pivot is too small compared to numbers below it it will pick the largest and reorder to put the largest pivot up there. So you like large pivots. Why do you like large pivots? Because for numerical stability that gives you small multipliers. If this is the biggest number then all the multiples that you need, the multiple of that row that you want to subtract from the other rows will be less than one. The multiplier is, so just let's remember, the multiplier is the entry to eliminate divided by the pivot. So if that entry to be eliminated is smaller than the pivot, then the multiplier's less than one and things stay stable. So that's a picture of elimination, a sort of first picture. And if it's sparse that means there's a lot of

zeroes. And where would we like those zeroes to be? Where would we like to find zeroes to save time and use to avoid all the steps, to avoid going through every step. If this was already a zero then it's there and we don't need to get it there. The job is already done. We can go forward. And if we have a zero here then it's already zero in the next one and we don't have to do, that would be the second pivot row, we don't have to subtract it from this row because a zero is already there.

So we like zeroes. And sparse matrices have a lot of them. But we only like them if they're in the right place. And what's the right place? It's, roughly speaking, it's the beginning of a row. So we're very happy with, we would like to get an ordering that looks somehow like this. We would like to get a bunch of zeroes in there. Well that's, I really overdid it. Maybe we can't get that many. But my point is, you see the point that those are at the beginning of the rows. So let's say we can't quite get to that, but maybe something more like that. Zero. So we get some rows starting with zeroes and then we get rows that don't. Which of course, I mean, they were there in the first place. What we've done is move them to the bottom. And the reason for moving them to the bottom is that they don't destroy rows below them. Yeah.

I didn't show how that would happen. Suppose I had a number there. Suppose that wasn't a zero. And all these were. Then when I do that elimination I subtract a multiple of that row from this row. Do you see that those zeroes will fill in? So fill in is the bad thing that you're trying to avoid. So in this quick discussion of reordering you're trying to order the rows so that they start with zeroes for as long as you can because those zeroes will never fill in. Zeroes inside a band, yes, just to make the point again, suppose I have a long row. Then if I have some zeroes here, here, here, useless. Because when I go to do elimination on these rows, when I subtract a multiple of that row from that row, that zero will fill in, it didn't help, it's gone. And this and this. So zeroes sort of near the diagonal are often going to fill in. It's zeroes at the far left that are good.

So that's a discussion of a topic that actually comes into 18.086. And so does the third words that I thought I would just write on the board. So this would be topic specializing in how do you solve large systems. And reordering with elimination is one way. And a second approach is, I just put these words up here so that you've seen them. Conjugate gradient method, that's a giant success in solving symmetric problems. Large symmetric problems. Something called multigrid is a terrific, incomplete LU. People have worked hard on ideas for solving problems so large that this becomes too expensive.

But backslash is the natural choice. If you can do it, it's like it's so simple. So let me focus on backslash. So backslash is the key to talk about. Let me see, let's just think what does backslash do? Suppose I had two equations, two different right-hand sides, b and c with the same matrix A . Would I solve them, would I do $u=A\backslash b$ and then separately $A\backslash c$? No. If I had two equations with the same matrix, there's a big saving in, suppose I have two equations, two, well, can I put it? I'll make a matrix. I mean this is what MATLAB would always do. Put those two right-hand sides into a matrix. Then backslash is happy with that. That backslash command would solve both equations. It would give you the answer then, would be, it would be two parts to it. What would be the first column of u then? So, do you see what I have now? I have a square matrix A times an unknown u . And I've got two right-hand sides. The point is this often happens.

If you're doing, say, a design problem. You might try different designs. So those different designs give you different right-hand sides. And then you have a problem to solve. What's the response to those designs? What are the displacements, what happens? So you want to solve with two right-hand sides and the point is you don't want to go through the elimination process on A twice. That's crazy. You see how that elimination process on A has this, what I described here, didn't look at b. I didn't even get to the right-hand side part. I was dealing with the expensive part, which is the matrix A. So we don't want to pay that price twice. And therefore backslash is all set up. So what's the first column of u then? It's $A \setminus b$. And now what's another way to write, a more mathematical way to write A? What's the answer to $Au=b$. It's $u = A^{-1}b$, right? That's the answer in the first column and $A^{-1}c$ would be the answer in the second column. So it would produce those answers. Backslash will produce the answer to the first equation and the answer the second equation. Will it do that by finding A^{-1} ? No. A^{-1} , for multiple reasons, we don't often compute, if it's 3 by 3, 4 by 4, then it's not a bad idea to see what A^{-1} looks like. We'll do that actually, because it's very enlightening if the matrix is small, you can see what's going on. But for a large problem we don't want A^{-1} . We want the answer. And backslash goes to the answer. It doesn't get there by computing A^{-1} . And our examples will show how.

So I'm giving A^{-1} a little bad comments right now. So I maybe should finish that sentence and then you'll see me turn around and compute the darn thing. But why do we not use A^{-1} ? Two reasons. One is it's more expensive. If I have to compute A^{-1} and multiply by b, that's taking too long. Second reason is A^{-1} could easily be a full, dense matrix. All non-zero. Where A itself was like, tridiagonal. So if A is tridiagonal, all the numbers we need are there in three diagonals, we don't want A^{-1} , you'll see, A^{-1} is full. So two reasons for not using A^{-1} . Takes too long in the first place even in the good case and often our matrix A has got lots of zeroes that are not there in A^{-1} so we've wasted time.

Nevertheless let me say something about A^{-1} on the next board. I don't know if you ever thought about the inverse matrix. Let me ask you this question. Suppose I use the command $A \setminus I$. What's that doing? That's putting the identity on the right-hand side instead of a single vector b or instead of two vectors b and c I'm now putting n. That's ok. Backslash will work with that. That's a shorthand for solving. This solves all these equations. A, it'll get my different answers, u_1 , u_2 . It'll get n different answers from the n right-hand sides, the columns of I. Let me take to be three. 0, 1, 0; 0, 0, 1. But I'll take n to be three. That's the identity. So this solves that equation. $A \setminus I$ will output u, will output u. And here's the question. What have I got? If you have a matrix A, square matrix and of course you have to create I as $\text{eye}(3)$, that I would be, this would be $\text{eye}(3)$. Cleve Moler's lousy pun. So what would I get? What would I get? The inverse, yes. I'd get the inverse. I'd get the inverse matrix. A backslash-- why's that? Because what matrix solves, what's the solution u to A times something equal I? The solution to this equation is $Au=I$. The solution to that equation is the inverse matrix. u will be A^{-1} . So that is a pretty, I mean, that's pretty, if I wanted A^{-1} that's a good way to do it. Other ways to do it would be $\text{inv}(A)$ in MATLAB and other ways, but this is about as good as you get. Do you see that you get the inverse matrix that way?

And it's worth giving some words to that fact. How would I describe this? This is a set of three different problems. I would describe, that right-hand side as an impulse. That's an impulse. A delta vector with an impulse in the first component. I'd call that

an impulse in the second component. I'd call that an impulse in the third component. So my inputs are three impulses and my outputs are u_1 , u_2 , u_3 . What words might I use? I could call those impulse response. If I were in Course 6, I certainly would. These would be impulses, these would be the responses to that impulse from our system. So those are impulse responses but in linear algebra the words I would use, u_1 , u_2 , u_3 are the columns of A inverse, right? That's what we just said. u_1 , u_2 , u_3 . So the columns of-- Let me write that down. The columns of A inverse are the responses, the u 's, the solutions to the impulses, to n impulses. And these are the columns of I . Do you see? Nothing I've said so far is deep or anything. But it's just, this comes up so much that it's nice to have different ways to think about this.

And actually, yeah, if I had to solve by hand, if I had to find the inverse by hand I would use elimination on this system of equations. I would take A and put the identity next to it. I would do elimination a lot. Actually I'll put that in one. If I had to find the inverse I would take this block matrix. Is that the first block matrix we've seen? Block matrices are really, you should just get familiar, when you're getting familiar with matrices, they often come in blocks. So here's a three by three block, here's a three by three block, the whole matrix is three by six. I can go through the elimination steps on it. If I really go haywire on elimination and keep going and going and going all the way until A gets to the identity, so I do elimination and I get it triangular then I even clean out up above the pivots and then I change all the pivots to one, I can get all the way to the identity if the matrix is invertible. And what do you think will show up in the right half? A inverse, yeah. So A inverse will show up there. So in 18.06 I would explain, go through examples of this computation just to see A inverse appear. There's no reason for us to go through long examples like that. M one three by three would be worth doing, but in the big picture we're going to use backslash.

Questions or discussion. So this is the sort of overall picture about the inverse. Well this is about the inverse. This was about elimination. I've gotta take a little time on elimination. No it's just too important. It's sort of straightforward and mechanical but it's like, too important to blow away. So let me remove that and put something better there. So now I'm talking about elimination. I'm talking about one equation. I'm back to $Au=b$. $Au=b$. And notice I'm using the letter A rather than K or one of our special letters because right now I don't know that that's a special matrix. In a minute the example I do will be one of our special matrices of course, and then I'll use it's letter. Just a word, though. I thought I would take-- you're getting a lot of big picture here for a minute and then we'll get into the details. I thought I would just, like, this is an occasion to look ahead. To say a word about the big picture of linear algebra. It's got four major problems, linear algebra. And there are four commands to solve those problems. And those commands, why not know? So those commands are LU. I'm speaking about MABLAB notation but Octave, Scilab, Python, R, all other would have those. Would do these same things.

So LU is the command that produces this, but I didn't say what that is yet. Ooh. So that's my job, right. What does this mean? What's up there? So, okay, to nobody's surprise MATLAB thought, ok, LU was a good letter for that. And what did MATLAB think of as a good letter for the command that does this? QR. So if I did $lu(A)$ or $qr(A)$ I would get-- I mean, this is sometimes associated with the names of Gram-Schmidt. It makes vectors orthogonal. Not to worry about this stuff. You can like, close your eyes for a moment here. lu is the first command and that's what today's about. qr is the key command for least squares problems. Maybe the biggest

application of rectangular matrices, I'm sure that's the big. Eigenvalues, do you know about eigenvalues? Well we'll just name the command $\text{eig}(A)$. And the singular value D composition, which you may never have heard of, but you will, is $\text{svd}(A)$. Can I leave that? It's in the videotape now. And my point is that when we've spoken about those four, we really have got numerical linear algebra and a lot of pure linear algebra explained. These are the four big ones.

And I guess what I'm saying is, the four big problems of linear algebra turn out to, a good way to describe the answer is as a factorization of the matrix. This is a factor. So now let me say what this one is. I start with a matrix and what elimination is really doing, if you look to see what is it doing, it's producing a lower triangular times an upper triangular. Let's go directly to that. Let me go directly to that. Let me take an example. So here's my matrix. Well, I don't have to call it A because you recognize it. So what's our name for that matrix? T. T because the top boundary condition is free. Oh, that reminds me. Some good comments after class Friday brought out something that I sloughed over. That the free-fixed matrix, the free-fixed problem is usually one unknown larger than the fixed-fixed. Because remember the fixed-fixed problem had both ends fixed. They were not unknowns. The only unknowns were one, two, three to n in the middle. But people noticed when I was talking about the free boundary condition that u_0 came into it and u_0 is not known. So really, the free boundary condition like, has an extra unknown, an extra row and column in the matrix and that's correct. We'll see later in Fourier transforms cosine matrices are one size bigger than sine matrices. The cosine matrices are free-free and the sine matrices are fixed-fixed. And now here we're at free-fixed.

I want to do elimination on that matrix. And while I'm at it, we'll find the inverse. But let's do elimination. Just on that matrix. Just to see what this L and U stuff is. What do we do? The first pivot is? One, it's fine. Not going to worry about that. We'll use it now. So how do I use it? I use a pivot, now listen because here is a convention here, I'm going to use the word subtract. You would say add that row to that row, right? Because you want to get a zero here. Forgive me for making it sound harder. I'm going to say subtract because I like subtraction. Subtract minus one of that row, my multiplier is minus one here. I'm going to say subtract minus one of that row from that. Same thing. You'll say okay. No problem. Let's just do it. So there's the pivot row and now when I-- shall I just add? When I add that to or does my superego thing subtract minus one of that from that. What do I get? I get the zero, the one and the minus one. And then what do I get? What's the multiplier? So let's just put these L, these multipliers, the l_{21} . That's the multiplier. 2, 1 refers to row two, column one. And this step got the zero in row two, column one. And what was the multiplier that did it? It's the number that I multiplied row one by and subtracted from row two, so it was minus one. What's l_{31} ? What's the multiplier that produces a zero in the three, row three, column one position? It's zero. I take zero of this row away from this row because it's zero already. So I'm not going to change, that row won't change and l_{31} was zero. Now I know the next pivot. I'm ready to use it. I want to get a zero below it because I'm aiming at this upper triangular u. And what's the multiplier now? And what's its number? What's the multiplier number? 3, 2 because I'm trying to fix row three, column two. And what do I multiply this by and subtract from this to make it zero? It's negative one again. Negative one, right. It's negative one. Sorry. Right. And now what happens when I do that? Can I just do it in place here? Forgive me if I just add that to that. And I'll get zero and one.

And now what do I know at this point, what have I learned? The most important thing I've learned is the matrix is invertible. Because the pivots one, one, and one,

well they're all here on the diagonal. This is my matrix U. That's my upper triangular matrix. And-- yeah, of course? I'm subtracting from this, so I've got the two is there, yeah, yeah, that's right. So the two is sitting there and I'm subtracting minus one of that row from it and that would mean taking, yeah, yeah. Right. That would mean I'm subtracting one from the two and getting the one. Yeah, yeah. So the row, the typical entry is, the typical result is the row you have minus L times the pivot row. The row you have minus the multiplier times the pivot row. That's the operation that elimination lives on. Elimination does that all the time. It's one of the basic linear algebra subroutines. B L A S.

Now, this is my U. So that's the goal of elimination, get upper triangular. And the reason is, you can solve upper triangular systems really fast. These multipliers l_{21} , l_{32} , and so on, they go into the L matrix. And now, let me just say it here that in a way, that example is too beautiful. Seldom am I sorry to see an example come out beautifully, but why do I say this is too beautiful? It's not typical. If I had other numbers here, I would get to other numbers here and what would be the difference, typically? The pivots wouldn't be all ones. That's what's too beautiful here, but let's go with it. I mean, it was worth it because everything came out simple. But the pivots for another problem, ooh, let me just do a second problem here. I'll do the fixed-fixed guy. Ok, so let's just do elimination on that. That's the first pivot. Subtract. Now what's the multiplier now? You're not as quick as MATLAB. MATLAB is ahead of you. So the multiplier is negative $1/2$. So the multiplier is, l_{21} is negative $1/2$. l_{31} will again be zero and let's use it, so it knocks that guys out. And what did that number come out to be? Do you remember? That was $3/2$. I think we looked at that once. And that would be all the same. And then the next multiplier, l_{32} will be negative $2/3$ because when I multiply that by $2/3$ it gives me the negative one and then I subtract and it kills this and I get $4/3$. I just did that quickly. And my main point was the pivots are on the diagonal. They're not all ones now. So this is a more typical one. This is, again our u. And our L matrix will be, oh, oh, that's the point. That these l's, these multipliers fit right into a lower triangular matrix. All these multipliers, and we'll put ones on the diagonal of that guy and these lower triangular ones will fit in just right perfectly in there. Over here the L would be, let me construct the L. Ones on the diagonal representing the pivot rows that stayed put and minus one, zero, and minus one as the multipliers that, so this was L, the multipliers that we used.

One more. We're doing lots of good stuff here and it's not deep, but it's-- Suppose the matrix had been singular. We have to realize, okay, this elimination method is great. But it can break down and it's going to break down, it has to break down somehow if the matrix is singular. Now what's our example of a singular matrix here? The matrix, this is free-fixed and that by fixing one support it wasn't singular, but if I want to make it singular, what'll I take? Free-free. Free-free matrix. So can I, if I had thought to bring colored chalk, I'll just erase for a moment for the bad case. The bad case would be free-free. And how would it show up as bad in elimination. How does a singular matrix reveal itself as elimination goes forward? Because you can't tell at the beginning. What would have gone wrong? We would have had a zero there. We had a two that dropped to one. But if we start with a one, it'll drop to zero. That would have been a zero there. The matrix would not have had three pivots. This upper triangular matrix is singular, no good. And that tells us back there that the original matrix is singular, no good. So if I can't get to three pivots somehow, the matrix'll be singular and that's an example that is. And MATLAB would immediately tell us, of course.

So let's go back to the good case for the main point. The good case for the main point. So the good case was three pivots. In fact it was extra good because they all turned out to be ones. Now, oh, now we're ready for LU. Here's the magic. And I'm not giving a proof. The magic is that the result U, if I multiply the multiplier matrix L times the result U, I'll bring back A. I'll bring back A. So let me just see. If I multiply L by U, so this is now L times U, maybe you can see that I get A. So what is U? I just have to copy it. $[1, 1, 1; -1, -1, 0]$. I could fill in the zeroes but I know they're there. That's L times U. And sure enough, if I do the multiplication, this-- How would you do that multiplication? I would say this is one of the first row when I see that. 1, 0, multiplying these. I'd say get one of the first row. That's correct in A. Here I would say this is minus one of the first row, plus one of the second row, and sure enough it's the right part of A. And this is minus one of the second row, plus one of the third row, and sure enough it's the right third row of A. I get A. And that's when elimination goes through with no zero pivots, no problems, just a bunch of multipliers, then that wonderful description of it, $A=LU$ is correct. I don't know how many that's new to. I should maybe have thought ahead. How many have seen like, L times U before? Just to give me an idea? Quite a few. Ok. So it's terrific. Oh, here I would get L times U. Now this is like a little more interesting because the pivots were not ones. So that's my matrix U. And here's my matrix L, right?

Okay, big point. Because we're so interested in symmetric matrices and this one in particular, or that one, symmetric matrices are good. Now, I'm unhappy about one aspect. So now there's just one part of this. This was great, we got three non-zero pivots, we got to U, we got the multiplier matrix all fine and we would be ready for the right-hand side and we would be ready for two right-hand sides, we would even be ready for all three right-hand sides, whatever. But I have one criticism. The matrix A which was our K, this was really K, was symmetric. That was the very first thing you did, told me on the very first day. And now it's equal to L times U, but what's happened? The symmetry is lost. Somehow the L has ones on the diagonal, the U as we have it has pivots on the diagonal, now the pivots are not all ones. So you see the symmetry of the problem got lost, and that shouldn't happen. And there ought to be a way to get back. Ok.

And now I want to describe the way to get back to symmetry. So LU doesn't keep the symmetry. L has ones, U has pivots. Different. But a very simple idea will bring back the symmetry. That is peel off the pivots into a diagonal matrix. In other words, there's a matrix, I'll call it D, D for diagonal. I'll divide those numbers out of each row. And can I just do that? So I'm just going to write this U as a product of this diagonal D where I'm going to be dividing the two out. So when I divide the two out from that row I'm left with one, minus 1/2, zero. And when I divide 3/2, the pivot then, it makes that pivot into a one and what does it produce for that guy? When I divide 3/2, when I divide that minus one by 3/2, what do I get? I get negative, division will be 2/3, I'll get a negative 2/3. And now, on the last row I'm dividing that row by 4/3. When I divide that row by 4/3, what row do I get here? . Because I've made the pivots one, well they're not pivots. What I've done is separate out the pivots. So I've made the diagonal ones just by separating it out. And what's happened? My goal was to get back some symmetry that was there at the start. Now so I have a pivot matrix D, and what's that matrix? You could say, well, it's the rest. But that's not what I'm looking for. What is it? Can everybody have a look at it? I can't raise it. If you look at what we got there, what is it? What's the right name to give it?

L transpose, exactly! That's the right name. L transpose. So what am I concluding then? I'm concluding that, let's see, where shall I put this? And it'll come back to it. Well, here we had just to show it wasn't an accident, here we had L, L transpose and what was the pivot matrix in this too beautiful problem case? It was the identity. So we didn't notice it. So can I squeeze in the identity? That's the pivot matrix there. But and again, we had L times L transpose. The beauty was there, the symmetry was there. And now what's the usual thing?

So really I'm completing this to one more thought. In that case when A is symmetric. I'm completing, I have the L. I'm factoring out the D. And what's left is L transpose. I hope you like LD^*L transpose. Seeing a matrix on one side and the transpose on the other side, the matrix L at the left and L transpose at the right is just right. So the point of symmetric case we have, and I'll use the letter K rather than A because now we're getting the matrix that's more special. It's that K or it's this T or it's any other symmetric matrix. The elimination leads to, uses multipliers L and if I factor out the pivot matrix then the other part is L transpose. We've seen that just by example. By two examples.

Now I want to just look at that. Because this that describes not only the result of elimination which is the key operation, but it also keeps the symmetry. In fact every matrix of that sort is symmetric. No, yeah, that's important. This is sure to be symmetric. We will often see matrices multiplied by their transpose. So what I'm saying is that if you gave me any matrix L, any diagonal matrix D, and then the transpose of L, if I multiplied those out, I would get a symmetric matrix. And going the other way, if I started with a symmetric matrix and I did elimination and got an L, then the D factoring out would leave me L transpose. So what you've seen by example is what will happen all the time.

Now why is that matrix symmetric? Here we get a chance to show the power of matrix notation, really. I just think that if I have any matrix L, in this case it happened to be lower triangular, but if I have any matrix L and I have a nice, symmetric diagonal guy in the middle and I have the transpose of this matrix on the other side I think the result is a symmetric matrix when I multiply. So it's these symmetric factorizations that we're getting to and are important problems because our important problems are symmetric. Ok, why is that sure to be symmetric? Suppose I asked you as an exercise, prove that L times a diagonal times L transpose is always a symmetric matrix. How could you do that? How could you do that? You could certainly create an example that did it and check it out, multiply, it would work. But we want to see that this is going to be true always. So how would you do that? I guess, let me get started. I would take its transpose. If I want to show something's symmetric, I transpose it and see if I get the same matrix again. So let me take the transpose of this. So I'm answering, Why is it sure to be symmetric? So let me take K transpose. So this is the transpose of, I have K equals something times something times something, A times B times C, you could say. If I transpose a matrix, how can I create transposes out of a, B and C. Do you remember what happens? They reverse the order.

It's like inverses. Transposes and inverses both have that key rule. When you have a product and you invert it, they come in the opposite order. When you transpose it, they come in the opposite order. So let me try put these separate transposes in the opposite order. So I've used the most important fact there. Which is just a fact about transposing a product. Ok, what have I got now? What's L transpose transposed? It's L, great. What's L transpose L transposed? Nothing but L. Transpose twice and I'm

back to L. What about D transpose? Same as D. Because D was symmetric, in fact diagonal. So what have I learned? The proof is done. I've got K back again. This was the original K. So I've learned that K transpose is K. So you're going to see time after time, let me just put these things there, you're going to see an A transpose times an A. That's the most important, most highly important multiplication. Take a matrix, maybe rectangular, multiply by A transpose. So this matrix is certainly square because A could be m by n. And then A transpose would be n by m and the result would be n by n. So it's certainly square. But now what's the new property we now know? It's symmetric. It's symmetric because if I transpose it, the transpose of A will go on this side, the double transpose will go on this side, but the double transpose is A again, so symmetric.

So I'm plugging away here on symmetric matrices because they're just-- yeah, what does symmetry mean in, yeah, can I just come back to this idea of responses? And by the way, if this was symmetric, would its inverse be symmetric? The answer is yes. If a matrix is symmetric, its inverse is symmetric. These symmetric matrices are a fantastic family. So I could add that to this. K inverse will also symmetric without having yet said why. But maybe in words, I'll just say a few words here at the end.

So what's a typical entry? Say the 2, 1 entry, just to carry on with this language one more moment. This is A inverse here. Now the 2, 1 entry in the inverse is an impulse is in the first, the first mass, whatever gets an impulse. And that is the response of the second mass. The response in position two to the impulse in position one. Now my matrix is symmetric, thinking about symmetric matrices here. So what about here? Here, if I take the impulse in position two and look at the response in position one, so do you see the difference? In general, those could be different. This is the response at position two to an impulse at one. This is the response at one to an impulse at two. You see that I'm multiplying those columns and getting these columns. And what's the point about symmetry? Those are the same. Symmetry is expressing this physical meaning that the response that i, to an impulse at j is the same as the response at j to an impulse at i. And that's sort of, that's such an important property, you want to notice it. And it goes into symmetry. So many, many problems will be symmetric and then some won't. We'll have to admit this won't cover everything, but it covers such an important and beautiful part.

So that's today's lecture on LU elimination, solving linear systems, and then let's move forward to understanding the actual inverses.