PROFESSOR: Let's go. So if you want to know the subject of today's class, it's A x = b. I got started writing down different possibilities for A x = b, and I got carried away. It just appears all over the place for different sizes, different ranks, different situations, nearly singular, not nearly singular. And the question is, what do you do in each case?

So can I outline my little two pages of notes here, and then pick on one or two of these topics to develop today, and a little more on Friday about Gram-Schmidt? So I won't do much, if any, of Gram-Schmidt today, but I will do the others. So the problem is A x = b. That problem has come from somewhere. We have to produce some kind of an answer, x.

So I'm going from good to bad or easy to difficult in this list. Well, except for number 0, which is an answer in all cases, using the pseudo inverse that I introduced last time. So that deals with 0 eigenvalues and zero singular values by saying their inverse is also 0, which is kind of wild. So we'll come back to the meaning of the pseudo inverse.

But now, I want to get real, here, about different situations. So number 1 is the good, normal case, when a person has a square matrix of reasonable size, reasonable condition, a condition number-- oh, the condition number, I should call it sigma 1 over sigma n. It's the ratio of the largest to the smallest singular value. And let's say that's within reason, not more than 1,000 or something.

Then normal, ordinary elimination is going to work, and Matlab-- the command that would produce the answer is just backslash. So this is the normal case. Now, the cases that follow have problems of some kind, and I guess I'm hoping that this is a sort of useful dictionary of what to do for you and me both.

So we have this case here, where we have too many equations. So that's a pretty normal case, and we'll think mostly of solving by least squares, which leads us to the normal equation. So this is standard, happens all the time in statistics. And I'm thinking in the reasonable case,

that would be ex hat. The solution A-- this matrix would be invertible and reasonable size.

So backslash would still solve that problem. Backslash doesn't require a square matrix to give you an answer. So that's the good case, where the matrix is not too big, so it's not unreasonable to form a transpose. Now, here's the other extreme. What's exciting for us is this is the underdetermined case. I don't have enough equations, so I have to put something more in to get a specific answer.

And what makes it exciting for us is that that's typical of deep learning. There are so many weights in a deep neural network that the weights would be the unknowns. Of course, it wouldn't be necessarily linear. It wouldn't be linear, but still the idea's the same that we have many solutions, and we have to pick one. Or we have to pick an algorithm, and then it will find one.

So we could pick the minimum norm solution, the shortest solution. That would be an L2 answer. Or we could go to L1. And the big question that, I think, might be settled in 2018 is, does deep learning and the iteration from stochastic gradient descent that we'll see pretty soon-- does it go to the minimum L1? Does it pick out an L1 solution?

That's really an exciting math question. For a long time, it was standard to say that these deep learning AI codes are fantastic, but what are they doing? We don't know all the interior, but we-- when I say we, I don't mean I. Other people are getting there, and I'm going to tell you as much as I can about it when we get there.

So those are pretty standard cases. m = n, m greater than n, m less than n, but not crazy. Now, the second board will have more difficult problems. Usually, because they're nearly singular in some way, the columns are nearly dependent. So that would be the columns in bad condition. You just picked a terrible basis, or nature did, or somehow you got a matrix A whose columns are virtually dependent-- almost linearly dependent.

The inverse matrix is really big, but it exists. Then that's when you go in, and you fix the columns. You orthogonalize columns. Instead of accepting the columns A1, A2, up to An of the given matrix, you go in, and you find orthonormal vectors in that column space and orthonormal basis Q1 to Qn. And the two are connected by Gram-Schmidt.

And the famous matrix statement of Gram-Schmidt is here are the columns of A. Here are the columns of Q, and there's a triangular matrix that connects the two. So that is the central topic

of Gram-Schmidt in that idea of orthogonalizing. It just appears everywhere. It appears all over course 6 in many, many situations with different names. So that, I'm sort of saving a little bit until next time, and let me tell you why.

Because just the organization of Gram-Schmidt is interesting. So Gram-Schmidt, you could do the normal way. So that's what I teach in 18.06. Just take every column as it comes. Subtract off projections onto their previous stuff. Get it orthogonal to the previous guys. Normalize it to be a unit vector. Then you've got that column. Go on.

So I say that again, and then I'll say it again two days from now. So Gram-Schmidt, the idea is you take the columns-- you say the second orthogonal vector, Q2, will be some combination of columns 1 and 2, orthogonal to the first. Lots to do. And there's another order, which is really the better order to do Gram-Schmidt, and it allows you to do column pivoting.

So this is my topic for next time, to see Gram-Schmidt more carefully. Column pivoting means the columns might not come in a good order, so you allow yourself to reorder them. We know that you have to do that for elimination. In elimination, it would be rows. So elimination, we would have the matrix A, and we take the first row as the first pivot row, and then the second row, and then the third row.

But if the pivot is too small, then reorder the rows. So it's row ordering that comes up in elimination. And Matlab just systematically says, OK, that's the pivot that's coming up. The third pivot comes up out of the third row. But Matlab says look down that whole third column for a better pivot, a bigger pivot. Switch to a row exchange. So there are lots of permutations then.

You end up with something there that permutes the rows, and then that gets factored into LU. So I'm saying something about elimination that's just sort of a side comment that you would never do elimination without considering the possibility of row exchanges. And then this is Gram-Schmidt orthogonalization. So this is the LU world. Here is the QR world, and here, it happens to be columns that you're permuting.

So that's coming. This is section 2.2, now. But there's more. 2.2 has quite a bit in it, including number 0, the pseudo inverse, and including some of these things. Actually, this will be also in 2.2. And maybe this is what I'm saying more about today. So I'll put a little star for today, here. What do you do?

So this is a case where the matrix is nearly singular. You're in danger. It's inverse is going to be big-- unreasonably big. And I wrote inverse problems there, because inverse problem is a type of problem with an application that you often need to solve or that engineering and science have to solve. So I'll just say a little more about that, but that's a typical application in which you're near singular. Your matrix isn't good enough to invert.

Well, of course, you could always say, well, I'll just use the pseudo inverse, but numerically, that's like cheating. You've got to get in there and do something about it. So inverse problems would be examples. Actually, as I write that, I think that would be a topic that I should add to the list of potential topics for a three week project. Look up a book on inverse problems.

So what do I mean by an inverse problem? I'll just finish this thought. What's an inverse problem? Typically, you know about a system, say a network, RLC network, and you give it a voltage or current. You give it an input, and you find the output. You find out what current flows, what the voltages are. But inverse problems are-- suppose you know the response to different voltages.

What was the network? You see the problem? Let me say it again. Discover what the network is from its outputs. So that turns out to typically be a problem that gives nearly singular matrices. That's a difficult problem. A lot of nearby networks would give virtually the same output.

So you have a matrix that's nearly singular. It's got singular values very close to 0. What do you do then? Well, the world of inverse problems thinks of adding a penalty term, some kind of a penalty term. When I minimize this thing just by itself, in the usual way, A transpose, it has a giant inverse. The matrix A is badly conditioned. It takes vectors almost to 0.

So that A transpose has got a giant inverse, and you're at risk of losing everything to round off. So this is the solution. You could call it a cheap solution, but everybody uses it. So I won't put that word on videotape. But that sort of resolves the problem, but then the question-- it shifts the problem, anyway, to what number-- what should be the penalty? How much should you penalize it?

You see, by adding that, you're going to make it invertible. And if you make this bigger, and bigger, and bigger, it's more and more well-conditioned. It resolves the trouble, here. And like today, I'm going to do more with that. So with that, I'll stop there and pick it up after saying something about 6 and 7. I hope this is helpful.

It was helpful to me, certainly, to see all these possibilities and to write down what the symptom is. It's like a linear equation doctor. Like you look for the symptoms, and then you propose something at CVS that works or doesn't work. But you do something about it. So when the problem is too big-- up to now, the problems have not been giant out of core.

But now, when it's too big-- maybe it's still in core but really big-- then this is in 2.1. So that's to come back to. The word I could have written in here, if I was just going to write one word, would be iteration. Iterative methods, meaning you take a step like-- the conjugate radiant method is the hero of iterative methods.

And then that name I erased is Krylov, and there are other names associated with iterative methods. So that's the section that we passed over just to get rolling, but we'll come back to. So then that one, you never get the exact answer, but you get closer and closer. If the iterative method is successful, like conjugate gradients, you get pretty close, pretty fast.

And then you say, OK, I'll take it. And then finally, way too big, like nowhere. You're not in core. Just your matrix-- you just have a giant, giant problem, which, of course, is happening these days. And then one way to do it is your matrix. You can't even look at the matrix A, much less A transpose. A transpose would be unthinkable. You couldn't do it in a year.

So randomized linear algebra has popped up, and the idea there, which we'll see, is to use probability to sample the matrix and work with your samples. So if the matrix is way too big, but not too crazy, so to speak, then you could sample the columns and the rows, and get an answer from the sample. See, if I sample the columns of a matrix, I'm getting-- so what does sampling mean? Let me just complete this, say, add a little to this thought.

Sample a matrix. So I have a giant matrix A. It might be sparse, of course. I didn't distinguish over their sparse things. That would be another thing. So if I just take random X's, more than one, but not the full n dimensions, those will give me random guys in the column space. And if the matrix is reasonable, it won't take too many to have a pretty reasonable idea of what that column space is like, and with it's the right hand side.

So this world of randomized linear algebra has grown because it had to. And of course, any statement can never say for sure you're going to get the right answer, but using the inequalities of probability, you can often say that the chance of being way off is less than 1 in 2 to the 20th or something. So the answer is, in reality, you get a good answer.

That is the end of this chapter, 2.4. So this is all chapter 2, really. The iterative method's in 2.1. Most of this is in 2.2. Big is 2.3, and then really big is randomized in 2.4. So now, where are we? You were going to let me know or not if this is useful to see. But you sort of see what are real life problems. And of course, we're highly, especially interested in getting to the deep learning examples, which are underdetermined.

Then when you're underdetermined, you've got many solutions, and the question is, which one is a good one? And in deep learning, I just can't resist saying another word. So there are many solutions. What to do? Well, you pick some algorithm, like steepest descent, which is going to find a solution. So you hope it's a good one. And what does a good one mean verses a not good one? They're all solutions.

A good one means that when you apply it to the test data that you haven't yet seen, it gives good results on the test data. The solution has learned something from the training data, and it works on the test data. So that's the big question in deep learning. How does it happen that you, by doing gradient descent or whatever algorithm-- how does that algorithm bias the solution?

It's called implicit bias. How does that algorithm bias a solution toward a solution that generalizes, that works on test data? And you can think of algorithms which would approach a solution that did not work on test data. So that's what you want to stay away from. You want the ones that work. So there's very deep math questions there, which are kind of new. They didn't arise until they did.

And we'll try to save some of what's being understood. Can I focus now on, for probably the rest of today, this case, when the matrix is nearly singular? So you could apply elimination, but it would give a poor result. So one solution is the SVD. I haven't even mentioned the SVD, here, as an algorithm, but of course, it is.

The SVD gives you an answer. Boy, where should that have gone? Well, the space over here, the SVD. So that produces-- you have A = U sigma V transposed, and then A inverse is V sigma inverse U transposed. So we're in the case, here. We're talking about number 5. Nearly singular, where sigma has some very small, singular values.

Then sigma inverse has some very big singular values. So you're really in wild territory here with very big inverses. So that would be one way to do it. But this is a way to regularize the

problem. So let's just pay attention to that. So suppose I minimize the sum of A x minus b squared and delta squared times the size of x squared. And I'm going to use the L2 norm.

It's going to be a least squares with penalty, so of course, it's the L2 norm here, too. Suppose I solve that for a delta. For some, I have to choose a positive delta. And when I choose a positive delta, then I have a solvable problem. Even if this goes to 0, or A does crazy things, this is going to keep me away from singular.

In fact, what equation does that lead to? So that's a least squares problem with an extra penalty term. So it would come, I suppose. Let's see, if I write the equations A delta I, x equals b 0, maybe that is the least squares equation-- the usual, normal equation-- for this augmented system. Because what's the error here? This is the new big A-- A star, let's say.

X equals-- this is the new b. So if I apply least squares to that, what do I do? I minimize the sum of squares. So least squares would minimize A x minus b squared. That would be from the first components. And delta squared x squared from the last component, which is exactly what we said we were doing. So in a way, this is the equation that the penalty method is solving.

And one question, naturally, is, what should delta be? Well, that question's beyond us, today. It's a balance of what you can believe, and how much noise is in the system, and everything. That choice of delta-- what we could ask is a math question. What happens as delta goes to 0? So suppose I solve this problem. Let's see, I could write it differently.

What would be the equation, here? This part would give us the A transpose, and then this part would give us just the identity, x equals A transpose b, I think. Wouldn't that be? So really, I've written here-- what that is is A star transpose A star. This is least squares on this gives that equation. So all of those are equivalent. All of those would be equivalent statements of what the penalized problem is that you're solving.

And then the question is, as delta goes to 0, what happens? Of course, something. When delta goes to 0, you're falling off the cliff. Something quite different is suddenly going to happen, there. Maybe we could even understand this question with a 1 by 1 matrix. I think this section starts with a 1 by 1. Suppose A is just a number.

Maybe I'll just put that on this board, here. Suppose A is just a number. So what am I going to call that number? Just 1 by 1. Let me call it sigma, because it's certainly the leading singular

value. So what's my equation that I'm solving? A transpose A would be sigma squared plus delta squared, 1 by 1, x-- should I give some subscript here? I should, really, to do it right.

This is the solution for a given delta. So that solution will exist. Fine. This matrix is certainly invertible. That's positive semidefinite, at least. That's positive semidefinite, and then what about delta squared I? It is positive definite, of course. It's just the identity with a factor. So this is a positive definite matrix. I certainly have a solution.

And let me keep going on this 1 by 1 case. This would be A transpose. A is just a sigma. I think it's just sigma b. So A is 1 by 1, and there are two cases, here-- Sigma bigger than 0, or sigma equals 0. And in either case, I just want to know what's the limit. So the answer x-- let me just take the right hand side. Well, that's fine.

Am I computing OK? Using the penalize thing on a 1 by 1 problem, which you could say is a little bit small-- so solving this equation or equivalently minimizing this, so here, I'm finding the minimum of-- A was sigma x minus b squared plus delta squared x squared. You see it's just 1 by 1? Just a number.

And I'm hoping that calculus will agree with linear algebra here, that if I find the minimum of this-- so let me write it out. Sigma squared x squared and delta squared x squared, and then minus 2 sigma xb, and then plus b squared. And now, I'm going to find the minimum, which means I'd set the derivative to 0. So I get 2 sigma squared and 2 delta squared.

I get a two here, and this gives me the x derivative as 2 sigma b. So I get a 2 there, and I'm OK. I just cancel both 2s, and that's the equation. So I can solve that equation. X is sigma over sigma squared plus delta squared b. So it's really that quantity. I want to let delta go to 0. So again, what am I doing here? I'm taking a 1 by 1 example just to see what happens in the limit as delta goes to 0.

What happens? So I just have to look at that. What is the limit of that thing in a circle, as delta goes to 0? So I'm finding out for a 1 by 1 problem what a penalized least squares problem, ridge regression, all over the place-- what happens? So what happens to that number as delta goes to 0?

1 over sigma. So now, let delta go to 0. So that approaches 1 over sigma, because delta disappears. Sigma over sigma squared, 1 over sigma. So it approaches the inverse, but what's the other possibility, here? The other possibility is that sigma is 0. I didn't say whether

this matrix, this 1 by 1 matrix, was invertible or not. If sigma is not 0, then I go to 1 over sigma.

If sigma is really small, it will take a while. Delta will have to get small, small, small, even compared to sigma, until finally, that term goes away, and I just have 1 over sigma. But what if sigma is 0? Sorry to get excited about 0. Who would get excited about 0? So this is the case when this is 1 over sigma, if sigma is positive. And what does it approach if sigma is 0? 0!

Because this is 0, the whole problem was like disappeared, here. The sigma was 0. Here is a sigma. So anyway, if sigma is 0, then I'm getting 0 all the time. But I have a decent problem, because the delta squared is there. I have a decent problem until the last minute. My problem falls apart. Delta goes to 0, and I have a 0 equals 0 problem. I'm lost.

But the point is the penalty kept me positive. It kept me with his delta squared term until the last critical moment. It kept me positive even if that was 0. If that is 0, and this is 0, I still have something here. I still have a problem to solve. And what's the limit then? So 1 over sigma if sigma is positive. And what's the answer if sigma is not positive? It's 0.

Just tell me. I'm getting 0. I get 0 all the way, and I get 0 in the limit. And now, let me just ask, what have I got here? What is this sudden bifurcation? Do I recognize this? The inverse in the limit as delta goes to 0 is either 1 over sigma, if that makes sense, or it's 0, which is not like 1 over sigma. 1 over sigma-- as sigma goes to 0, this thing is getting bigger and bigger.

But at sigma equals 0, it's 0. You see, that's a really strange kind of a limit. Now, it would be over there. What have I found here, in this limit? Say it again, because that was exactly right. The pseudo inverse. So this system-- choose delta greater than 0, then delta going to 0. The solution goes to the pseudo inverse. That's the key fact.

When delta is really, really small, then this behaves in a pretty crazy way. If delta is really, really small, then sigma is bigger, or it's 0. If it's bigger, you go this way. If it's 0, you go that way. So that's the message, and this is penalized. These squares, as the penalty gets smaller and smaller, approaches the correct answer, the always correct answer, with that sudden split between 0 and not 0 that we associate with the pseudo inverse.

Of course, in a practical case, you're trying to find the resistances and inductions in a circuit by trying the circuit, and looking at the output b, and figuring out what input. So the unknown x is the unknown system parameters. Not the voltage and current, but the resistance, and inductance, and capacitance.

I've only proved that in the 1 by 1 case. You may say that's not much of a proof. In the 1 by 1 case, we can see it happen in front of our eyes. So really, a step I haven't taken here is to complete that to any matrix A. So that the statement then. That's the statement. So that's the statement. For any matrix A, this matrix, A transpose A plus delta squared inverse times A transpose-- that's the solution matrix to our problem.

That's what I wrote down up there. I take the inverse and pop it over there. That approaches A plus, the pseudo inverse. And that's what we just checked for 1 by 1. For 1 by 1, this was sigma over sigma squared plus delta squared. And it went either to 1 over sigma or to 0. It split in the limit. It shows that limits can be delicate.

The limit-- as delta goes to 0, this thing is suddenly discontinuous. It's this number that is growing, and then suddenly, at 0, it falls back to 0. Anyway, that would be the statement. Actually, statisticians discovered the pseudo inverse independently of the linear algebra history of it, because statisticians did exactly that. To regularize the problem, they introduced a penalty and worked with this matrix.

So statisticians were the first to think of that as a natural thing to do in a practical case-- add a penalty. So this is adding a penalty, but remember that we stayed with L2 norms, staying with L2, least squares. We could ask, what happens? Suppose the penalty is the L1 norm.

I'm not up to do this today. Suppose I minimize that. Maybe I'll do L2, but I'll do the penalty guy in the L1 norm. I'm certainly not an expert on that. Or you could even think just that power. So that would have a name. A statistician invented this. It's called the Lasso in the L1 norm, and it's a big deal.

Statisticians like the L1 norm, because it gives sparse solutions. It gives more genuine solutions without a whole lot of little components in the answer. So this was an important step. Let me just say again where we are in that big list. The two important ones that I haven't done yet are these iterative methods in 2.1.

So that's like conventional linear algebra, just how to deal with a big matrix, maybe with some special structure. That's what numerical linear algebra is all about. And then Gram-Schmidt with or without pivoting, which is a workhorse of numerical computing, and I think I better save that for next time. So this is the one I picked for this time.

And we saw what happened in L2. Well, we saw it for 1 by 1. Would you want to extend to

prove this for any A, going beyond 1 by 1? How would you prove such a thing for any A? I guess I'm not going to do it. It's too painful, but how would you do it? You would use the SVD. If you want to prove something about matrices, about any matrix, the SVD is the best thing you could have-- the best tool you could have.

I can write this in terms of the SVD. I just plug-in A equals whatever the SVD tells me to put in there. U sigma V transposed. Plug it in there, simplify it using the fact that these are orthogonal. If I have any good luck, it'll get an identity somewhere from there and an identity somewhere from there. And it will all simplify. It will all diagonalize.

That's what the SVD really does is turns my messy problem into a problem about their diagonal matrix, sigma in the middle. So I might as well put sigma in the middle. Yeah, why not? Before we give up on it-- a special case of that, but really, the genuine case would be when A is sigma. Sigma transpose sigma plus delta squared I inverse times sigma transpose approaches the pseudo inverse, sigma plus.

And the point is the matrix sigma here is diagonal. Oh, I'm practically there, actually. Why am I close to being able to read this off? Well, everything is diagonal here. Diagonal, diagonal, diagonal. And what's happening on those diagonal entries? So you had to take my word that when I plugged in the SVD, the U and the V got separated out to the far left and the far right. And it was that that stayed in the middle.

So it's really this is the heart of it. And say, well, that's diagonal matrix. So I'm just looking at what happens on each diagonal entry, and which problem is that? The question of what's happening on a typical diagonal entry of this thing is what question? The 1 by 1 case! The 1 by 1, because each entry in the diagonal is not even noticing the others.

So that's the logic, and it would be in the notes. Prove it first for 1 by 1, then secondly for diagonal. This, and finally with A's, and they're using the SVD with and U and V transposed to get out of the way and bring us back to here. So that's the theory, but really, I guess I'm thinking that far the most important message in today's lecture is in this list of different types of problems that appear and different ways to work with them.

And we haven't done Gram-Schmidt, and we haven't done iteration. So this chapter is a survey of-- well, more than a survey of what numerical linear algebra is about. And I haven't done random, yet. Sorry, that's coming, too. So three pieces are still to come, but let's take the last two minutes off and call it a day.