GILBERT STRANG: Well, so three things to mention. One was you remember last time I made a list of six or seven different situations where Ax equal b and problems that could arise? The last one was when A was just way too big to fit into core. But in the middle were other methods. So other issues, like the columns being nearly dependent when Gram-Schmidt will come up.

First, I want to say that I know there are typos in the two pages. I thought you might just like to see the very first draft of two pages of the book. People sometimes ask me, how long does the book take to write? So I started when 18.065 started a year ago. So I'm into the second year, and it usually takes two years.

And the system is I write by hand so. I wrote those opening pages that you saw, two pages, by hand. Then I scan them to Mumbai, where my best friend in the world types them with typos. No problem, because I'm going to make so many changes that a few typos are nothing.

Anyway, so he scans them back to me typed. And then I start making changes. If I'm lucky I'd have the chance to talk about it in here. Then I realize better things to do. Then I scan back to him and back to me and back to him and back to me. So that's where the two years disappear. Anyway, I'm quite happy with those two pages, until I start improving them.

And one other topic from the past that came out in class. It wasn't in the notes yet. Do you remember the day that we minimized different norms? L1 or L2 or max, L infinity norm with the condition of solving with a constraint that this equation was satisfied. I'm in 2D to be able to draw a picture. And the constraint is one line. And that's about what the line looks like.

So I'm going to draw here again-- what I'm doing is putting some numbers in to that insight I drew about a week ago. Do you remember that? Because I thought that really illustrates how L1 and L2 and L infinity are different.

Let me draw the L2 one here. So where's the point on this line-- so x has to lie on this line. And where is the point that has the smallest sum of squares norm, standard L2 norm? So

geometrically where is that point? Well, what does the set of points with norm 1 look like for L2? It's a circle, right.

So we just blow that circle up or shrink it down until it touches this thing. And where it touches, it'll touch where the radius is perpendicular. So there's our best point in L2, because if we picked another point, the norm would have to be bigger to go through that point. So that's clearly the first one. And actually, we can probably see what it is, because if we know those are perpendicular, I know the slope of this line. so I think that the slope of this line is something like 3/4, probably coming from there, or maybe 4/3. We'll figure it out. I think it's 4/3. I'll find that point.

Then the most interesting one was the L1 norm, because what was the shape of the unit ball for L1?

AUDIENCE: A diamond.

GILBERT STRANG: A diamond, right. A diamond. So the diamond that first touches the line is here. That's the winning point. And if the line is 3x1 plus 4x2 equal 1, then I know that that point is-- x1 will be 0 and x2 will be 1/4. So that's the winning point in L1.

And I think I calculated this right that the winning point in L2 I think will be-- let's see, this goes up to-- I'm moving down the line. I think this would have 3/25, 4/25. I won't stop to derive that. But at least-- yeah, the slope is looking like 4/3. It goes up 4 when it crosses 3. And the 4 and the 3 came from there. And, of course, you notice that I've scaled it to fit the line. 3 times 3/25 is 9/25 plus 16/25, 25/25 is 1.

And finally, what about L infinity? What was the picture there? What's the unit ball look like in 2D for the max norm? It's a--

AUDIENCE: Square.

GILBERT STRANG: Square, right. So the square will hit there. These two on the square are-- this is a 45 degree line now on that square. It hits it at that sharp point. And so the x1 and x2 are equal. And I think they're probably-- let's see, if they're equal, if we made them 1/7, then I'd have 3/7 plus 4/7 equals 7/7. Yeah, I think we make some 1/7, 1/7. So that would be the x infinity point. 3/7 plus 4/7 give 1.

So why do I mention this? First, because when I did it before I just drew pictures without really

solving the problem. And secondly, because in thinking ahead about projects, this is the kind of project that I would think is quite interesting. Obviously, as this is p equal 1. This is p equal 2. This is p equal infinity. And I guess it's pretty clear from the pictures that as p increases, this point starts up here. And moves down the line and ends up here. And I've no idea like what the solution is for a different p and how it moves.

And to make it more of a project, what happens in 3D? What happens in 3D? In 3D, if I have one equation in 3D, then I have a plane. And this would become a diamond, a 3D diamond. This would be a 3D sphere. This would be a 3D cube. They would expand to hit that plane. And I don't know how many zeros you get in that case. So that would be the case of one equation. So there would be a plane that these diamond, sphere, and cube expand until they hit it.

Or you could have two constraints, two equations. So if we had two equations and three unknowns, that would be a line again. But how many zeros would we get in these different cases? How sparse is L1 going to be?

That's like a recapture of what we did. It's nice occasionally to have pictures showing where the solution is. Now, I'm coming to the topic of the day, which is Gram-Schmidt. And so Gram-Schmidt, number one, is the standard way that would be taught in 18.06.

So what's Gram-Schmidt about? I'll just put down here general facts of Gram-Schmidt. We start with a matrix A. It's got n columns. But they're not orthogonal. And, in fact, they may be badly conditioned. Columns might be nearly dependent on the others. I'm going to assume the columns are independent, but they might be barely independent. So those lines then would be sort of like pointing very nearly parallel.

But Gram-Schmidt opens up the picture to get a matrix Q, an orthogonal matrix with columns Q1 to Qn, which our ortho-normal. So it gets a perfect basis of Qs. And so that's what Gram-Schmidt does. And these are different ways to do it, different ways to organize the computation. I really only put the standard way in.

What is this mysterious R? So the Qs are combinations of the As. So there's some matrix to tell me what those combinations are. Or if I go back go backwards and say, well, the As are combinations of the Qs, that's what I'm about to do. If I say each A is a combination of Qs, that means that my A matrix is my Q matrix times some R matrix, which tells me the combinations. When I multiply by R on the right, I'm taking combinations of the columns of Q and getting the

columns of A.

So just like LU, you go forward with the algorithm to reach U. Here, we go forward where the algorithm to reach Q. But then when we want to put it in one simple equation, it turns out to be better to go backwards and say how is the original A related to the final Q, there has to be some R.

OK, I always feel when I talk about Gram-Schmidt-- I usually end with that A equal QR. And, of course, the Matlab command is exactly QR. So in Matlab, the command would be QR of A, instead of LU of A. So it would give you Q and R. That's what Matlab will output.

Now, as I say, Q is the saying we're constructing. R is the combinations that we need to get what we want. And so it comes at the end, what the heck was R. But actually, R is really a simple idea. So I want to show that at the beginning instead of the end.

OK, so I'm going to move Q over here, as Q inverse. But what is Q inverse?

AUDIENCE: Q transpose.

GILBERT STRANG: Q transpose, right, because I've created an orthogonal matrix. So this mysterious R is Q transpose A. And let me just sort of make it grow out into matrices. That has the Qs along the rows, Q transpose, of course. Qn transpose, Q1 transpose. I'm transposing the matrix above it. So these columns become rows. Times the As, A1 to An.

So what is a typical entry in R? That's really why I want to say nothing mysterious about this. You can see what you end up with. It will be right in front of us here. What is the entry in row i, column j of R?

OK, this says that all those entries in R are Qi transpose times Aj. That's the old way to multiply matrices. And it's the best way for this, a row times a column. In other words, the Rs are just the inner products, the dot products of the Qs with the As, of the Qs with the As. That's sort of like nothing mysterious about R. Because Q is an orthogonal matrix, we were able to put it over here, get a nice expression for R, and see what it really is. So you can do R at the end or on the root. But that's just the inner product of the Qs with the A's.

Now, what's Gram-Schmidt? I'm sort of thinking you've seen the basic ideas of Gram-Schmidt, but let's review. So I start with a. So what does Gram-Schmidt begin with? a1. It takes that first column. So these a's are not orthogonal generally. But the first direction is OK. I have no

complaints about the first direction, except that a1 might not be a unit vector. So q1 will just be a1 over its norm to have a unit vector.

The whole idea of Gram-Schmidt is in q2. So what is q2? The whole idea is coming here. It's the only thing you need to know. And the picture shows it.

So q2, I start with a2. But it's not orthogonal to a1. So what do I do? I figure out the component of a2 in the a1 direction and I remove it. So I take that vector away and I'm left with this vector. So there is a vector. I'll call that A2. So A2 is the original little a2 with the a1 direction removed.

So what what's the formula for what I just did? This is the whole, the key step that Gram-Schmidt repeats over and over and over again. It's truly boring.

So it subtracts-- well, remember that this is in the same direction as Q1. And it's better to work with Q1, because we've found that guy. We've got it. And we know it's a unit vector.

So here's my linear algebra question. What's the component of a2 that I want to subtract off? It's the component in the direction of q1. It's this in the direction of q1. And let me just remember, so obviously, that angle is coming into it. So that will be a2 transpose q1 times q1. That's it. That's the component that we remove.

And maybe I'd prefer to write it as q1 transpose a2. I don't know. It doesn't matter of course. The two dot products are the same. Maybe I will just-- yeah-- well, maybe not. Fine.

Now what is that vector supposed to achieve? It's supposed to be this vector. This vector I'm really going to call A2, because it's in the right direction for Q2, but it is not yet?

AUDIENCE: Normal.

GILBERT STRANG: Normal. So what is Q2 then? So I'm saying this guy got the direction right. They're saying subtract it off this vector. Got that direction right. Got it as A2.

What is Q2 now that I want to finish? I've got the direction. All I want to do is get it to be a unit vector. So I just take A2 over its norm.

That double step is the whole thing in Gram-Schmidt, the whole thing. Subtract off the components in the directions already set. Then you get something in a totally new direction, called A, capital A. And then you divide by its length to make it a unit vector. And that gives you the new Q.

Just to show that we've got a point, what about the next step, aiming for Q3. So tell me what A3 should be? A3, I'm going to start with this. And I'm going to subtract off some stuff. What am I going to subtract off?

AUDIENCE: The transpose.

GILBERT STRANG: The component, a3 transpose, right. Times q1 q1. And I didn't yet check so that this came out orthogonal to q1, but I'll come back to that. Now, have I done everything I should do here with a3?

No, I've got one more step to take. And I should take the two steps separately. It's called modified Gram-Schmidt. And what I want to do is subtract off the q2 component. So what multiple of q2 do I need? Because q2 has been set by the time I get to a3, so what goes here?

AUDIENCE: a3 transpose

GILBERT STRANG: a3 transpose--

AUDIENCE: q2

GILBERT STRANG: q2. Thanks. Thanks. If you look at a code, say a Matlab code to do Gram-Schmidt-- oh, what's the final q3 then?

AUDIENCE: Normalize it.

GILBERT STRANG: Normalize it. So you take A3, which is in the right direction, and you divide by its length to get a unit vector.

Let me just come back and check that I did it right, that I got the right direction. So what do I mean by the right direction? What should I check about that guy? I should check that it's inner product with q1 is? If this is the right direction to go, that way, then I should check-- I have to check-- hopefully, I've got the formula right-- that it's dot product, inner product with q1 is--

AUDIENCE: 0.

GILBERT STRANG: 0. Thank you. So is it obvious that it is? Take the inner product of the dot product of that with q1, what do you get? You get the same number q1 a2 transpose q1. You get that number. And over here, you're getting q1 transpose with q1. So do you see what I'm doing? Probably it

looks like it would have been better.

I'm checking that q1 transpose a2 is 0. Yeah, it is. q1 transpose a2 here, q1 transpose a2, or a2 transpose q1, I don't mind. And I have another q1 transpose q1 here. And what is that? What is q1 transpose q1? It is?

AUDIENCE: 1.

GILBERT STRANG: 1. So check. OK, that's Gram-Schmidt, standard Gram-Schmidt, which you have met before. Now, I'm ready for a better Gram-Schmidt. You could say a better Gram-Schmidt, because here-- so what's it going to be the difference? Here, I took the a's in their original order.

Now, suppose I did that with elimination. Elimination, usually we write as acting on the row. So thinking about elimination, I'm thinking I'm the rows. What would be the danger in taking the rows in order, doing no row exchanges, just figure out the pivot each time, and kill the rest of the column and then move on? So taking the rows in the order they came, whatever it might be, what's the risk? And why would Matlab not do that?

Because something can be very small and totally blow up your calculations. And that's that pivot number. Sort of the question is, if a2 is very near a1-- so let me draw a new picture. So here's the risk. So a1 was whatever it was. If a2 is really close to the same direction, then I'm subtracting off almost all of it, and I've got some tiny little bit for the new direction.

That's like the pivot in elimination. It's the number that sort of measures what's new, what the new row in elimination or the new column in Gram-Schmidt gives you. And if that's too small-- I mean, like in elimination, as I say, we would never use an elimination code on a general matrix that didn't check the size of the pivot and exchange rows when necessary.

Well, similarly, with Gram-Schmidt, it can take the columns in order. That's the standard Gram-Schmidt taking the columns in order. But only if it checks each time that the little bit-- well, that the new part, what would be the new part, is big enough to be able to-- we have to divide by the thing. And if that A2 is tiny is a tiny vector. This is dividing by A2 and onwards is building in round-off error that we can't remove again. We're stuck with it.

So that's the column exchange, column pivoting idea in sort of a more professional Gram-Schmidt. And to do it-- so I have to be able to compare this little bit, if it is little, with what? What am I going to compare with? In elimination, I looked down the rest of the column for a

bigger number.

I guess what I have to do is I have to find this component, not just from a2, but from all the remaining a's. And I'll pick the biggest. So there, in that sentence, I said the main idea of column exchangers is once you get q1 set, which certainly q1 was the easiest one in the world, but maybe even for q1 I guess it could even happen. That would be like starting with a zero up in the upper left corner for elimination, like what a way to start the day.

Here, if my matrix A had a tiny little a1, then I should look for a bigger one to get the very first q chosen. Let's suppose-- give ourselves a reasonable chance here-- let's suppose a1 was a decent size, so as I've drawn. The next step might not be, if I use a2, as I say, it could be same direction as a1 virtually, and then I'm just working with that little piece. So what do I have to do differently? I have to be able to compare this little piece with all the other potential possibilities.

So let me just write down what you have to do in a different order. So this is now with column pivoting, column exchange, column pivoting allowed, or it's possible. So to make it possible, I have to find not only A2, the piece of little a2, I have to find a2, the piece. I'm just going to copy that. I have to take my second column, subtract off the q1 part. And that could be small.

So I have to compare it with-- oh, I haven't written this page up. So I haven't got a notation in mind yet. I won't give it a name. I have to also compute at this step before deciding q2-- now I'm describing how to decide q2, the second vector. And I'm saying that the way to decide q2 is not only to take a piece of a2 but also the piece of a3. Look at this piece. And look at all the other pieces.

And now, what will be my policy? Standard Gram-Schmidt accepted this one, and didn't look at these. But, now, I'm going to look at them all. And I'm going to take the largest. I'm going to take the largest. And that will be the A2 that I want.

So it might not be this one. If this guy is largest, then I'm taking column 3 first. And that will be my A2. And then I'll say fine. And then q2 will be that A2 over its norm. You see the difference? It's not exciting.

And you might think, wait a minute, this is a heck of a lot more work. But it isn't. It isn't actually more work, because these are all the things-- these ones that look like we're paying a price, we're computing all these alternatives-- but we had to do that eventually anyway. Do you see

that?

Let me just say it again. The standard way took all the components, like for here. The standard way waited until you got to column 3 and then subtracted off both pieces, waited until you got to column 4, subtracted off three pieces. This way, you're subtracting off the first piece as soon as you know what it should be. As soon as you know q1, you remove it from all the remaining vectors.

And you look to see what's the biggest. You pick the biggest one. I said maybe it's this guy. So you move that one. Or some permutation matrix is going to move it to the second column. See, it started in the third column, but I'm going to move it to the second, because it's the biggest. Then I do the right thing. I find q2. And now I go on toward q3.

And how will I find q3? I'd want to pick the biggest column to work with. So I subtract off the q2 components. This is like easy to say, but I had never like figured it out before. So let me just say it again. And then I'll leave that with you.

How do I get q3? I've fixed two columns. They happened to be, maybe not necessarily the first two, but two columns, two q's are set, and I'm looking for the next one.

I go on and I look at all the remaining columns, all of which have had subtracted off their q1 and q2 parts. So I've orthogonalized with respect to q1 and q2. I look at all the remaining things that I have to work with and pick the biggest. Just like picking the biggest number to go into the pivot. OK, I don't think I can say it anymore without just repeating myself. And I bring it here to class, because I had not sort of appreciated the point that no extra work was involved. You just did these subtractions for all the remaining columns of A before you started on the next job.

Is that OK? Eventually the notes will describe that. Maybe they even do. Yeah, I think they even do. I wrote it but I didn't understand it. Now, little improvement. So yes?

AUDIENCE: So are we permuting every time to get the biggest pivot?

GILBERT STRANG: Yeah. Yeah. Only we don't call them pivots. Or maybe we should. I don't know what word is used to get the biggest column remaining or something. Yeah, yeah, each time. You know, if the columns were in a stupid order, this puts them in the right order.

OK, finally, come these weird names, Krylov, Russian. Arnoldi, actually, I don't know what he

is. And I shouldn't admit that on tape. So what's the idea there? So again, we're solving Ax equal b. So this is going to be Krylov. What was his idea? Well, I want to solve Ax equal b.

A is a big matrix, pretty big. Of course, I don't plan to invert it. That would be insane. What I can do with a the matrix A, especially if it's sparse-- so a large sparse A would be a good candidate for Krylov.

So what is it that you could do cheap and fast with a large, I mean really large, but really sparse matrix A?

AUDIENCE: Matrix times a vector.

GILBERT STRANG: You can do a matrix times a vector. And here's our matrix. And there is our vector. So we could start with a vector b. We can multiply A times b. We can multiply A times A times b. And, of course, I write it that way. I never-- I mean, like if you multiply A times A first, then like you turn in your Matlab account, because you just have to do it that way.

And then you keep going, which of course is A squared b, but you didn't form A squared. And then on up to-- in the end, you get to some A to say k, k minus 1 b. But, of course, that's computed as A times the previous one, which was A times the previous one.

So there is a bunch of vectors, which are likely to be independent. So they span a space. And it's called the Krylov space. So these span.

They're combinations. Combinations give-- oh, I don't like that letter k, because that's also Krylov, so what shall I say? j. So I have j vectors. The original b, Ab, A squared b, up to that. So combinations give the Krylov space, say, we'll name it after Krylov and we need a subscript j to show how big it is, its dimension. So that will be the idea.

Well, let me complete the idea. The idea will be-- there are combinations. So that's a space, a subspace, pretty big if j is big. And I'm going to look for the best solution in that space. So I'm not going to solve of Ax equals b exactly. I'm going to find the best solution, the closest solution, the least squares solution in this Krylov space. I'm going to let j be pretty big. So this space has got plenty of vectors in it.

I have a basis for this space. And some combination of these basis vectors will be my xj. So, again, xj will be the best vector, or the closest vector in this Krylov space, j, It will be the best vector in that space, the closest one.

So I know what the space is. I've reduced the dimension down to j. And I can find this best vector.

There's just one catch. And it's the same catch that Gram-Schmidt were aiming to help to remove. That is this basis that I'm-- right now, I'm working with all combinations of these guys. And those could be very, very dependent. That might be a terrible basis. Anytime you want to do big computations, what kind of a basis do you want? Yes? So what sort of a basis is good to project onto to find the best solution within that subspace? So we're sort of finding a projection. And you've got vectors that span the space. So you know what you're projecting onto.

But those vectors, they might be nearly dependent. They might all be pointing almost the same direction. In which case, your calculations are terrible. So what do you do? Orth-- Orthogonalize. And that's where Arnoldi comes in. And there's also a Hungarian guy named Lanczos.

So that's what they contribute is how to orthogonalize that basis. And then, once you've done that, you have an orthogonal basis. And, of course, an orthogonal basis is perfect to do a projection. Everybody has to know that. Why is a orthogonal basis so great? Ortho-normal even. Let's just remember.

Suppose I have a vector x. It's unknown here. But suppose I have it. And I want to write it as a combination of these ortho-normal guys. say, n. What is it about all ortho-normal q's that makes this easy to do, which it would not be with an arbitrary basis? So this is really Q times c, right? Q times this vector of C's. The q's are in the columns of Q. The c's were multiplying a matrix by a vector. It's a combination of the columns. That's what we get.

And when the q's are orthogonal, what's the answer? We can get the answer straight away. So here, we're trying to find the coefficients with respect to the basis vectors Q of a given vector x.

And what's the answer to that question? The point is, usually, to find the coefficients, c would have to be Q inverse x. We'd have to solve that system of equations. We do have to solve that system of equations.

But where's the payoff from ortho-normal basis?

AUDIENCE: Q inverse--

GILBERT STRANG: Q inverse is Q transpose. That's the payoff. So it's just telling me that to find c1-- how do I find c1? This says take the first q1, transpose with x. I'll say the same thing here. Take the first vector with x. That will be about cq, q1, transpose q1. I'm just taking the dot product of everything there with q1. And then a c2, q1 transpose, q2 and so on.

But what's good?

AUDIENCE: You have zeros.

GILBERT STRANG: Tell me again.

AUDIENCE: The other ones are zeros.

GILBERT STRANG: These are all zero. And the q1 transpose q1 is?

AUDIENCE: 1.

GILBERT STRANG: 1. So it's perfect. c1 is q1 transpose x. And that's exactly what that tells us. The first component is the first row of q transpose, which is q1 transpose with x. So that's the idea. So that's the idea here. That's the reason for Arnoldi and Lanczos being famous is that they figured out a good way to orthogonalize that basis.

Do we want to see what they did? Or those would be in the notes. Well, how do you do it?

So this is the basis. This is our not good basis. And then our good basis is going to be q's. So I'll take b to be-- q1 will be what? What would be the right choice for q1? Well, I'll take that first vector and normalize it. We're just doing Gram-Schmidt.

What would q2 be? How would I find q2 following the Gram-Schmidt idea? I take Ab. I subtract off its component in this q1 direction and I normalize. And all the Arnoldi Lanczos algorithm is is that same Gram-Schmidt idea applied to these Krylov vectors.

So Arnoldi-Lanczos-- Arnoldi is for any matrix and Lanczos is for a symmetric matrix where you get some special benefit. So what they did, you could say now, they just wrote down Gram-Schmidt, in fact, probably the standard Gram-Schmidt, because this is a case where we

really don't want to exchange columns. I don't want suddenly to be pushed into this one. I'd rather take them in order, because it just turns out right. And this is in the notes. So let me tell you where this is. It will be Section II.1. So Part 2 of the book, which is where we are, and the first section.

So what all together is in this first section when you look at it? That section is standard numerical linear algebra, what any course in MIT offers, 18.3-- I'm not sure of the number, 330 maybe, which is, of all things like this. Krylov would be there, Arnoldi, Lanczos. Of course, Gram and Schmidt would be there. That's five people who've thought of the same thing.

And so that Section II.1 summarizes what's in really good, well, a lot of textbooks. And let me mention a favorite, a book by Trefethen and Bau. Or the "Bible." So this is maybe a moment to tell you about two books on classical numerical linear algebra, what do you do for matrices of order 1,000, not for matrices of order millions. That you have to rethink. So Trefethen-Bau isn't called numerical linear algebra.

And do you know the authors of the Bible of numerical linear algebra? So that's a textbook. And what I'm going to write down now finally is 750 pages it's grown to in its fourth edition. It's the Bible for all numerical linear algebra people. And it's written by Golub and VanLoan. So Gene Golub was a remarkable guy. He probably didn't write more than about 11 pages of this. Charlie wrote most of it.

But Golub was an amazing person who traveled the world and connected people and left behind papers to be written and books to be written. And so this Golub-VanLoan is now in the fourth volume. And it has so much good stuff and references that it's like the good reference. And this is the good textbook if you were going to teach a course on numerical linear algebra.

So I think that I've come to the point to finish. So I really have finished along with the extra attraction of this different problem, I finished with Ax equal b. And, well, at least, I now move onto what to do with really, really large matrices.