GILBERT STRANG: So let me use the mic to introduce Alex Townsend, who taught here at MIT-- taught Linear Algebra 18.06 very successfully.

And then now he's at Cornell on the faculty, still teaching very successfully. And he was invited here yesterday for a big event over in Engineering.

And he agreed to give a talk about a section of the book-- section 4.3-- which, if you look at it, you'll see is all about his work. And now you get to hear from the creator himself. OK.

ALEX TOWNSEND: OK. Thanks. Thank you, Gil. Thank you for inviting me here. I hope you're enjoying the course.

Today I want to tell you a little about why there so many matrices that are low rank in the world. So as computational mathematicians-- Gil and myself-- we come across low-rank matrices all the time. And we started wondering, as a community, why?

What is it about the problems that we are looking at? What makes low-rank matrices appear? And today I want to give you that story-- or at least an overview of that story.

So for this class, x is going to be n by n real matrix. So nice and square. And you already know, or are very comfortable with, the singular values of a matrix.

So the singular values of a matrix, as you know, are a sequence of numbers that are monotonically non-increasing that tell us all kinds of things about the matrix x.

For example, the number of nonzero singular values tell us the rank of the matrix x. And they also, you probably know, tell us how well a matrix x can be approximated by a low-rank matrix.

So let me just write two facts down that you already are familiar with. So here's a fact-- that, if I look at the number of non-zero singular values in x-- so I'm imagining there's going to be k non-zero singular values-- then we can say a few things about x.

For example, the rank of x, as we know, is k-- the number of non-zero singular values. But we also know from the SVD that we can decompose x into a sum of rank 1 matrices-- in fact, the sum of k of them.

So because x is rank k, we can write down a low-rank representation for x, and it involves k terms, like this.

Each one of these vectors here is a column vector. So if I draw this pictorially, this guy looks like this, right? And we have k of them. So because x is rank k, we can write x as a sum of k rank 1 matrices.

And we also have an initial fact that we already know-- that the dimension of the column space of x is equal to k, and the same with the row space. So the column space of x equals the row space of x-- the dimension-- and they all equal k.

And so there are three facts we can determine from looking at this sequence of singular values of a matrix x. Of course, the singular value sequence is unique. X defines its own singular values.

What we're interested in here is, what makes x? What are the properties of x that make sure that the singular values have a lot of zeros in that sequence? Can we try to understand what kind of x makes that happen?

And we really like matrices that have a lot of zeros here, for the following reason-- we say x is low rank if the following holds, right? Because if we wanted to send x to our friend-- we're imagining x as picture where each entry is a pixel of that image.

If that matrix-- that image-- was low rank, we could send the picture to our friend in two ways. We could send one every single entry of x. And for us to do that, we would have to send n squared pieces of information, because we'd have to send every entry.

But if x is sufficiently low rank, we could also send our friend the vectors-- u, u1, v1, uk, up to vk. And how much pieces of data would we have to send our friend to get x to them if we sent in the low-rank form?

Well, there's 2n here, 2n here numbers. There's k of them. So we'd have to send 2kn numbers. And we strictly say a matrix is low rank if it's more efficient to send x to our friend in low-rank form then in full-rank form.

So this, of course, by a little calculation, just shows us that, provided the rank is less than half the size of the matrix, we are calling the matrix low rank.

Now, often, in practice, we demand more. We demand that k is much smaller than this number, so that it's far more efficient to send our friend the matrix x in low-rank form than in full-rank form. So the colloquial use of the word low rank is kind of this situation. But this is the strict definition of it.

So what do low-rank matrices look like? And to do that, I have some pictures for you. I have some flags-- the world flags. So these are all matrices x-- these examples-- because their flags happen to not be square. I hope you can all see this.

But the top row here are all matrices that are extremely low rank. For example, the Austria flag-- if you want to send that to your friend, that matrix is of rank 1. So all you have to do is send your friend two vectors. You have to tell your friend the column space and the row space. And there's only the dimensions of one of both.

For the English flag, you need to send them two column vectors and two row vectors-- u1, v1, u2 and v2. And as we go down this row, they get slowly fuller and fuller rank.

So the Japanese flag, for example, is low rank but not that small. The Scottish flag is essentially full rank. So it's very inefficient to send your friend the Scottish flag in low-rank form. You're better off sending almost every single entry.

So what do low-rank matrices look like? Well, if the matrix is extremely low rank, like rank 1, then when you look at that matrix-- like here, like the flag-- it's highly aligned with the coordinates-- with the rows and columns.

So if it's rank 1, the matrix is highly aligned-- like the Austria flag. And of course, as we add in more and more rank here, the situation gets a bit blurry. For example, once we get into the medium rank situation, which is a circle, it's very hard to see that the circle is actually, in fact, low rank.

But what I'm going to do was try to understand why the Scottish flag or diagonal patterns-- particularly a bad example for low rank. So I'm going to take the triangular flag to examine that more carefully. So the triangular flag looks like-- I'll take a square matrix and I'll color in the bottom half.

So this matrix is the matrix of ones below the diagonal. And I'm interested in this matrix and, in particular, its singular values, to try to understand why diagonal patterns are not particularly useful for low-rank compression. And this matrix of all ones has a really nice property that, if I take its inverse, it looks a lot like-- getting close to Gil's favorite matrix.

So if I take the inverse of this matrix-- it has an inverse because it's got ones on the diagonal-- then its inverse is the following matrix, which people familiar with finite difference schemes will notice the familiarity between that and the first order finite difference approximation.

In particular, if I go a bit further and times two of these together, and do this, then this is essentially Gil's favorite matrix, except one entry happens to be different-- ends up being this matrix, which is very close to the second order, central, finite difference matrix.

And people have very well studied that matrix and know its eigenvalues, its singular values-- they know everything about that matrix. And you'll remember that if we know the eigenvalues of a matrix, like x transpose x, we know the singular values of x. So this allows us to show, by the fact that we know that, that the singular values of this matrix are not very amenable to low rank. They're all non-zero, and they don't even decay.

So I'm getting this from-- I rang up Gil, and Gil tells me these numbers. That allows us to work out exactly what the singular values of this matrix are, from the connection to finite differences.

And so we can understand why this is not good by looking at the singular values. So the first singular value of x from this expression is going to be approximately 2n over pi. And from this expression, again, for the last guy-- the last singular value of x is going to be approximately a half.

So these singular values are all large. They're not getting close to zero. If I plotted these singular values on a graph-- so here's the first singular value, the second, and the n-th-- then what would the graph look like?

Well, plot these numbers. Divide by this guy so that they all are bounded between 1 and 0 because of the normalization, because I divided by sigma 1 of x.

And so we can plot them, and they will look like this kind of thing. This number happens to be here where they come to be pi over 4n, which is me dividing this number by this number, approximately.

So triangular patterns are extremely bad for low rank. We need things-- or we at least intuitively think that we need things-- aligned with the rows and columns, but the circle case happens to also be low rank. And so what happened to the Japanese flag? Why is the Japanese flag convenient for low rank?

Well it's the fact that it's a circle, and there's lots of symmetry in a circle. So if I try to look at the rank of a circle, the Japanese flag, then I can bound this rank by decomposing the Japanese flag into two things. So this is going to be less than or equal to the rank of sum of two matrices, and I'll do it so that the decomposition works out.

I have the circle. I'm going to cut out a rank one piece that lives in the middle of this circle. OK? And I'm going to cut out a square from the interior of that circle. OK? And I can figure out-- of course the rank is just bounded by the sum of those two ranks. This guy is bounded by rank one because it's highly aligned with the grid. So this guy is bounded by rank one. So this thing here plus 1.

And now I have to try to understand the rank of this piece. Now this piece has lots of symmetry. For example, we know that the rank of that matrix is the dimension of the column space and the dimension of the row space. So when we look at this matrix, because of symmetry, if I divide this matrix in half along the columns, all the columns on the left appear on the right. So for example, the rank of this matrix is the same as the rank of that matrix because I didn't change the column space. OK?

Now I go again and divide along the rows, and now the row dimension of this matrix is the same as the top half, because as I wipe out those, I didn't change the dimension of the row space because the rows are the same top-bottom. And so this becomes the rank of that tiny little matrix there. And because it's small, it won't have too large a rank. So this is definitely less than-- if I divide that up, a little guy here looks like that plus the other guy that looks like that plus 1.

And so of course the row space of this matrix cannot be very high because this is a very thin matrix. There's lots of zeros in that matrix, only a few ones. And so you can go along and do a bit of trig to try to figure out how many rows are non-zero in this matrix.

And a bit of trig tells you-- well it depends on the radius of this original circle. So if I make the original radius r of this Japanese flag, then the bound that you end up getting will be, for this matrix, r 1 minus square root 2 over 2 for this guy. That's a bit of trig. I've got to make sure that's an integer. And then again, here it's the same but for the column space. So this is me just doing trig. OK?

And that's bound on the rank. It happens to be extremely good. And if you work out what that rank is and try to look back, you will find it's extremely efficient to send the Japanese flag to your friend in low rank form, because it's not full rank because these numbers are so small. So this comes out to be, like, approximately 1/2 r plus 1. So much smaller than what you would expect, because remember, a circle is almost the anti-version version of a line with the grid, but yet, it's still low rank. OK.

Now most matrices that we come up with in computational math are not exactly of finite rank. They are of numerical rank. And so I'll just define that. So the numerical rank of a matrix is very similar to the rank, except we allow ourselves a little bit of wiggle room when we define it, and so that amount of wiggle room will be of parameter called tol called epsilon. That's a tolerance. I'm thinking of epsilon as a tolerance. That's the amount of wiggle room I'm going to give myself. OK.

And we say that the numerical rank-- I'll put an epsilon there to denote numerical rank-- is k. k is the first singular value, or the last singular value, above epsilon. In the following sense, I'm copying the definition above but with epsilons instead of zeros. If this singular value is less than epsilon, relatively, and the kth one was not below. So k plus 1 is the first singular value below epsilon in this relative sense.

So of course the rank of 0x, if that was defined, is the same as the rank of x. OK? So this is just allowing ourselves some wiggle room. But this is actually what we're interested more in practice. All right? I don't want to necessarily send my friend the flag to exact precision. I would actually be happy to send my friend the flag up to 16 digits of precision, for example. They're not going to tell the difference between those two flags. And if I can get away with compressing the matrix a lot more once I have a little bit of wiggle room, that would be a good thing.

So we know from the Eckart and Young that the singular values tell us how well we can approximate x by a low-rank matrix. In particular, we know that the k plus 1 singular value of x tells us how well x can be approximated by a rank k matrix. OK? For example, when the rank was exactly k, the sigma k plus 1 was 0, and then this came out to be 0 and we found that x was exactly a rank k matrix.

Here, because we have the wiggle room, the epsilon, we get an approximation, not an exact. So this is telling us how well we can approximate x by a rank k matrix. OK? That's what the singular values are telling us. And so this allows us to try our best to compress matrices but use low-rank approximation rather than doing things exactly.

And of course, on a computer, when we're using floating point arithmetic, or on a computer because we always round numbers to the nearest 16-digit number, if epsilon was 16 digits, your computer wouldn't be able to tell the difference between x or x the rank k approximation if this number satisfied this expression. Your computer would think of x and xk as the same matrix because it would inevitably round both to epsilon, within epsilon. OK.

So what kind of matrices are numerically of low rank? Of course all low-rank matrices are numerically of low rank because the wiggle room can only help you but it's far more than that. There are many full-rank matrices-- matrices that don't have any singular values that are zero-- but the singular values decay rapidly to zero. That are full-rank matrices with low numerical rank because of the wiggle room.

So for example, here is the classic matrix that fits this regime. If I give you this, this is called the Hilbert matrix. This is a matrix that happens to have extremely low numerical rank but it's actually full rank, which means that I can approximate H by a rank k matrix where k is quite small very well, provided you give me some wiggle room, but it's not a low-rank matrix in the sense that if epsilon was zero here, you didn't allow me the wriggle room, all the singular values of this matrix are positive. So it's of low numerical rank but it's not a low-rank matrix.

The other classical example which motivated a lot of the research in this area was the Vandermonde matrix. So here is the Vandermonde matrix. An n by n version of it. Think of the xi's as real. And this is Vandermonde. This is the matrix that comes up when you try to do polynomial interpolation at real points.

This is an extremely bad matrix to deal with because it's numerically low rank, and often, you actually want to solve a linear system with this matrix. And numerical low rank implies that it's extremely hard to invert, so numerical low rank is not always good for you. OK? Often, we want the inverse, which exists, but it's difficult because V has low numerical rank.

OK. So people have been trying to understand why these matrices are numerically of low rank for a number of years, and the classic reason why there are so many low-rank matrices is because the world is smooth, as people say. They say, the world is smooth. That's why matrices are of numerical low rank. And to illustrate that point, I will do an example.

So this is classically understood by a man called Reade in 1983, and this is what his reason was. I have a picture of John Reade. He's not very famous, so I try to make sure his picture gets around. He's playing the piano. It's, like, one of the only pictures I could find of him.

So what is in this reason? Why do people say this? Well here's an example that illustrates it. If I take a polynomial in two variables and I-- for example, this is a polynomial of two variables-- and my x matrix comes from sampling that polynomial integers-- for example, this matrix-- then that matrix happens to be of low rank-- mathematically of low rank, with epsilon equals zero.

Why is that? Well if I write down x in terms of matrices, you could easily see it. So this is made up of a matrix of all ones plus a matrix of j-- so that's 1, 2, up to n, 1, 2, up to n, because every entry of that matrix just depends on the row index. And then this guy depends on both j and k. So this is a multiplication table, right? So this is n, 2, 4, up to 2n, n, 2n, n squared. OK.

Clearly, the matrix of all ones is a rank one matrix. The same with this guy. The column space is just of dimension one. And the last guy also happens to be of rank one because I can write this matrix in rank one form, which is a column vector times a row vector. OK. So this matrix x is of rank three. I guess at lowest rank three is what I've actually shown. OK.

Now of course this hasn't got to numerical low rank yet, so let's get ourselves there. So Reade knew this, and he said to himself, OK, well if I can approximate-- if x is actually coming from sampling a function, and I approximate that function by polynomial, then I'm going to get myself a low-rank approximation and get a bound on the numerical rank.

So in general, if I give you a polynomial of two variables, which can be written down-- it's degree n in both x and y. Let's just keep these indexes away from the matrix index. I give you this such polynomial, and I go away and I sample it and make a matrix X, then X, by looking at each term individually like I did there, will have low rank mathematically, with epsilon equals zero. This will have, at most, m squared rank, and if m is 3 or 4 or 10, it possibly could be low because this X could be a large matrix. OK.

So what Reade did for the Hilbert matrix was said, OK, well look at that guy. That guy looks like it's sampling a function. It looks like it's sampling the function 1 over x plus y minus 1. So he said to himself, well, that x, if I look at the Hilbert matrix, then that is sampling a function. It happens to not be a polynomial. It happens to be this function.

But that's OK because sampling polynomials, integers, gives me low rank exactly. Maybe sampling smooth functions, functions like this, can be well approximated by polynomials and therefore have low numerical rank. And that's what he did in this case. So he tried to find a p, a polynomial approximation to f. In particular, he looked at exactly this kind of approximation. So he has some numbers here so that things get dissolved later. And he tried to find a p that did this kind of approximation. So this approximates f.

And then he would develop a low-rank approximation to X by sampling p. So he would say, OK, well if I let y be a sampling of p, then from the fact that f is a good approximation to p, y is a good approximation to X. And so this has finite rank. He wrote down that this must hold. And the epsilon comes out here because these factors were chosen just right. The divide by n was chosen so that the epsilon came out just there. OK?

So, for many years, that was kind of the canonical reason that people would give, that, well, if the matrix X is sampled from a smooth function, then we can approximate our function by a polynomial and get polynomial rank approximations. And therefore, the matrix X will be of low numerical rank. There's an issue with this reasoning, especially for the Hilbert matrix, that it doesn't actually work that well.

So for example, if I take the 1,000 by 1,000 Hilbert matrix and I look at its rank-- OK, well I've already told you this is full rank. You'll get 1,000. All the singular values are positive. If I look at the numerical rank of this 1,000 by 1,000 Hilbert matrix and I compute it, I compute the SVD and I look at how many are above epsilon where epsilon is 10 to the minus 15, so that means I can approximate the 1,000 by 1,000 Hilbert matrix by a rank 28 matrix and only give up 15-- there will be exact 15 digits, which is a huge amount. So this is what we get in practice, but Reade's argument here shows that the rank of this matrix, the numerical rank, is at most. So it doesn't do a very good job on the Hilbert matrix for bounding the rank, right?

So Reade comes along, takes this function. He tries to find a polynomial that does this, where epsilon is 10 to the minus 15. He finds that the number of terms that he needs in this expression here is around 719, and therefore, that's the rank that he gets. The bound on the numerical rank. The trouble is that 719 tells us that this is not of low numerical rank, but we know it is, so it's an unsatisfactory reason.

So there's been several people trying to come up with more appropriate reasons that explain the 28 here. And so one reason that I've started to use is another slightly different way of looking at things, which is to say the world is Sylvester. Now Sylvester, what does that mean?

What does the word "Sylvester" mean in this case? It means that the matrices satisfy a certain type of equation called the Sylvester equation, and so the reason is really, many of these matrices satisfy a Sylvester equation, and that takes the form-- for sum A, B, and C.

OK. So X is your matrix of interest. You want to show X is of numerical low rank. And the task at hand is to find an A, B, and C so that X satisfies that equation. OK. For example, the two matrices I've had on the board satisfy a Sylvester equation-- a Sylvester matrix equation. There is an A, a B, and a C for which they do this.

For example, remember the Hilbert matrix, which we have there still, but I'll write it down again. Has these entries. So all we need to do is to try to figure out an A, a B, and then a C so that we can make it fit a Sylvester equation. There's many different ways of doing this. The one that I like is the following, where if I put 1/2 here and 3/2 here, all the way down to n minus 1/2, times this matrix-- so this is timesing the top of this matrix by 1/2 and then 3/2 and then 5/2. So we're basically timesing each entry of this matrix by j minus 1/2.

And then I do something on the right here, which I'm allowed to do because I've got the B freedom, and I choose this to be the same up to a minus sign. Then when you think about this, what is it doing? It's timing the jk entry-- this is-- by j minus 1/2. That's what this is doing. And what's this doing is timesing the jk entry by k minus 1/2. So this is, in total, timesing the jk entry by j plus k minus 1/2 minus 1/2, which is minus 1, so this is timesing the jk entry by j plus k minus 1. So it knocks out the denominator. And what we get from this equation is a bunch of ones. So in this case, A and B are diagonal, and C is the matrix of all ones. OK?

We can also do this for Vandermonde. So Vandermonde, you'll remember, looks like this. And then over here, we have this guy, the matrix that appears with polynomial interpolation. OK. So if I think about this, I could also come up with an A, B, and C, and for example, here's one that works. I can stick the x's on the diagonal.

So if you imagine what that matrix on the left is doing, it's timesing each column by the vector x. OK? So the first column of this matrix becomes x, the vector x. The second becomes the vector x squared, where squared is done entry-wise. And then the third entry is now x cubed, and when we get to the last, it's x to the n. OK? So that's like, multiply each column by the vector x.

So if I want to try to come up with a matrix-- so what's left is of low rank, is like of this form. What I can do is shift the columns. So I've noticed that this product here, this diagonal matrix, has made the first column x. So if I want to kill off that column, I can take the second column and permute it to the first column. I could take the third column and permute it to the second, the last column and permute it to the penultimate column here. And that will actually kill off a lot of what I've created in this matrix right here.

So let me write that down. This is a circumshift matrix. This does that permutation. I've put a minus 1 there. I could have put any number there. It doesn't make any difference. But this is the one that works out extremely nicely. Now this zeros out lots of things because of the way I've done the multiplication by x and the circumshift of the columns.

And so the first column is zero because this first column is x, this first column is x, so I've got x minus x. This column was x squared minus x squared, so I got zero, and I just keep going along until that last column. That last column is a problem because the last column of this guy is x to the n, whereas I don't have x to the n in V, so there are some numbers here. OK.

You'll notice that C in both cases happens to be a low-rank matrix. In these cases, it happens to be of rank one. And so people were wondering, maybe it's something to do with satisfying these kind of equations that makes these matrices that appear in practice numerically of low rank. And after a lot of work in this area, people have come up with a bound that demonstrates that these kind of equations are key to understanding numerical low rank.

So if X satisfies a Sylvester equation, like this, and A is normal, B is normal-- I don't really want to concentrate on those two conditions. It's a little bit academic. Then-- people have found a bound on the singular values of any matrix that satisfies this kind of expression, and they found this following bound. OK, so here, the rank of C is r. So that goes there. So in our cases, the two examples we have, r is 1, so we can forget about r.

This nasty guy here is called the Zolotarev number. E is a set that contains the eigenvalues of A, and F is a set that contains the eigenvalues of B. OK. Now it looks like we have gained absolutely nothing by this bound, because I've just told you singular values are bound by Zolotarev numbers. That doesn't mean anything to anyone. It means a little bit to me but not that much.

So the key to this bound-- the reason this is useful-- is that so many people have worked out what these Zolotarev numbers actually mean. OK? So these are two key people that worked out what this bound means. And we have gained a lot because people have been studying this number. This is, like, a number that people cared about from 1870 onwards to the present day, and people have studied this number extremely well. So we've gained something by turning it into a more abstract problem that people have thought about previously, and now we can go to the literature on Zolotarev numbers, whatever they are, and discover this whole literature of work on this Zolotarev number.

And the key part-- I'll just tell you the key-- is that the sets E and F are separated. So for example, in the Hilbert matrix, the eigenvalues of A can be read off the diagonal. What are they? They are between minus 1/2 and n minus 1/2. And the eigenvalues of B lie in the set minus 1/2 minus n plus 1/2. And the key reason why the Hilbert matrix is of low numerical rank is the fact that these two sets are separated, and that makes this Zolotarev number gets small extremely quickly with k.

Now you might wonder why there is a question mark on Penzl's name. There is an unofficial curse that's been going on for a while. Both these men died while working on the Zolotarev problem. They both died at the age of 31. One died by being hit by a train, Zolotarev. It's unclear whether he was suicidal or it was accidental.

Penzl died at the age of 31 in the Canadian mountains by an avalanche. I am currently not yet 31 but going to be 31 very soon, and I'm scared that I may join this list. OK. But for the Hilbert matrix, what you get from this analysis, based on these two peoples' work, is a bound on the numerical rank. And the rank that you get is, let's say, a world record bound. For the Hilbert matrix is 34, which is not quite 28, not yet, but it's far more descriptive of 28 than 719.

And so this technique of bounding singular values by using these Zolotarev numbers is starting to gain popularity because we can finally answer to ourselves why there are so many low-rank matrices that appear in computational math. And it's all based on two 31-year-olds that died. And so if you ever wonder when you're doing computational science when a low rank appears and the smoothness argument does not work for you, you might like to think about Zolotarev and the curse. OK, thank you very much.

[APPLAUSE]

GILBERT STRANG: Thank you [INAUDIBLE] Excellent.

ALEX How does it work now?

TOWNSEND:

GILBERT STRANG: We're good. Yeah.

ALEX TOWNSEND: I'm happy to take questions if we have a minute, if you have any questions.

GILBERT STRANG: How near of 31 are you?

ALEX TOWNSEND: [INAUDIBLE] I get a spotlight. I'm 31 in December.

GILBERT STRANG: Wow. OK.

ALEX TOWNSEND: So they died at the age of 31, so you know, next year is the scary year for me. So I'm not driving anywhere. I'm not leaving my house until I become 32.

GILBERT STRANG: Well, thank you [INAUDIBLE]

ALEX TOWNSEND: OK, thanks.

[APPLAUSE]