GILBERT STRANG: OK. Why don't I start? So I was hoping that we would have the next online assignment ready, but Julia 0.6, the new version, is slowing us down, and it'll probably be next time. But my lectures, of course, are-- well, what I want to say, perhaps, is this isn't intended to be a course in numerical linear algebra, but I thought I couldn't let the whole semester go by without saying something about how to compute eigenvalues and singular values.

Of course, you're going to call eig or SVD or the equivalent in Python or Julia. But actually, the QR factorization that we spoke about, that we spoke entirely about last time, is the key-- unexpectedly, unexpectedly. You have a matrix A whose eigenvalues you want. So let's start with eigenvalues. And it might be a symmetric matrix. I'll stay with A rather than S because it doesn't have to be, but you get special-- always you get something special if the matrix is symmetric.

So this method of computing eigenvalues, to me at least and I think to many people, came out of the blue a while ago, but not that long ago. And it worked very well. So here's the idea. It's called the QR method because you start by factoring your matrix into QR. So here's A. Can we call it A0? That's the matrix we start with whose eigenvalues we want. And I'll call these Q0 and R0.

And you remember what that means, what's hiding behind those letters that I've written? You have the columns of A, possibly symmetric as I said, but not orthogonal usually. So you find an orthogonal basis. You orthogonalize-- you line them up perpendicular to each other. And then there is a matrix R, which happens to be upper triangular, that connects the not-orthogonal basis with the orthogonality base. Right. We constructed R step-by-step.

And then, the idea is write these in the reverse order, and that will be A1, the next A. And then do it again and again and again. And so we're hoping, of course, that the eigenvalues didn't change. We're hoping that we can forget A0, start again with A1, and produce A2 and continue. So we're hoping two things. One will not be a hope. We can see that the eigenvalues of A1 and A0 are the same. So we have not changed the eigenvalues.

How do we see that? If you had to show that two matrices had the same eigenvalues, would you compute all the eigenvalues and compare them? Certainly not. What would you do? What's the best test, the usual test, the usual thing you would want to show to--

AUDIENCE: They're similar.

GILBERT STRANG: They're similar, that's right. So the claim would be that these two matrices are similar. Right. So maybe we should show that. So we want to write A1 in a different way. So the claim is A1 is similar to A. So we just have to figure out-- we have to get A1 and turn it to A0. So here is A1. So A1. I want to show that that's right. So that's R0, Q0.

But what is Q0? From here, Q0 is what we-- Q0 is-- what do I want? I want to put R0 inverse over there. So R0. Now for Q0, I'm going to substitute. So what is Q0? I multiply both sides of that equation by R0 inverse. So Q0 is A0 is-- sorry, I said A and wrote Q. A0, R0 inverse. For Q0, I've put in what it equals. And that's done it. That's exactly telling me that A1 which is this, equals this, and that's a similarity transformation. I have not changed the eigenvalues. So that's OK.

The other thing, which is the, you could say, the miracle in this thing, is that when I continue to do that for almost every matrix, the matrices begin to lose stuff off the diagonal, especially below the diagonal by the ordering that QR has done. So it would tend to-- you start with a matrix A0. You got a matrix A1, which is a little smaller here. This part being especially smaller.

You do it again. This is even smaller. Even smaller. A2. Even smaller. And you keep going. And for most matrices, the result is that-- I don't know how many steps we want to think of taking, but we get quite small numbers here, especially-- yeah. So once we get small numbers here, little epsilons-- that's everybody's shorthand for small numbers-- what would you expect to see on the diagonal?

AUDIENCE: The eigenvalues.

GILBERT STRANG: The eigenvalues, because this has the same eigenvalues as this, as this, as this, and these little epsilons are not going to change the eigenvalues too much. So these will be, on the diagonal, will be close to the eigenvalues. And actually, what happens is this one comes first. That one is quite accurate first. I guess we should probably do a simple example to see this happen.

Actually, I do an example in the notes. And let me say what happens. So I do a 2 by 2 example which has something like cos theta, sine theta. I don't know what-- I've forgotten what I took, and I don't have that page of notes here. Something here, something here, something here as A0. And, then A1, after just one step, has sine cubed theta there and numbers there that are getting much closer to the eigenvalues.

Sorry that this isn't a full-scale example, but the point of the example is here, that this off-diagonal entry gets cubed. And the next step will be its 9th power, and then the 27th power. So it's really quickly going to 0. And this happens-- so cubic convergence is-- that's a price in numerical linear algebra.

So that happens. So this arrived on the scene and quickly blew away all other methods that were being used to compute eigenvalues. But numerical people, being what they are, they wanted to improve it-- like, is there a way to make it faster? And it turned out there is. It turned out that the better way-- basically, the same idea, but just you're always looking for a simple change. And the idea of introducing a shift was tried and turned out to work extremely well. So that's the improvement.

So we take A1-- no. So how does it-- how does the shift work? So instead of A1-- have I got space here to do this? Instead of A0, I take A1 minus a shift. So a shift is some multiple of the identity. If I just move a matrix by a multiple of the identity, what happens to its eigenvectors? What happens to its eigenvalues? Something pretty simple, right, because I'm just shifting by sI. What happens to its eigenvectors?

AUDIENCE: They stay the same.

GILBERT STRANG: They're the same. And what happens to the eigenvalues?

AUDIENCE: They change by s.

GILBERT STRANG: They change by s. If A0 V equaled lambda v, then when I multiply this by V, there will be an extra-- it'll be lambda minus s because the identity times the V is just V. So it just shifts all the eigenvalues by s. And you try to shift-- you look for a shift. This would be great. If you knew lambda N, a shift there, you would be looking for 0 then if you shifted them all by that lambda. And it turns out that would speed things up.

So it will work instead with this matrix as now, again, I'm factoring it in a Q0 R0. So that's the work of the method is in doing Gram-Schmidt at every step. And then there's a little work in reversing the order. And then I want to undo the shift, so I do that factorization. Now, let's see. You may have to help me to remember what I should do here. So I factor those. I reverse those. And then I think I add back the shift, and that's my A1.

So I took A0. I shifted it. I worked with it. QR. Reversed the order. RQ. Added back the shift to get a matrix. And what am I, of course, hoping about the matrix A1 and A0? I'm hoping they're still similar. So I did a shift and I undid a shift. But of course, after doing our QR, I have to check, are these really still similar? So let me just try to check that one again. Maybe I'll just-- it's sitting right there, so let me do it again.

I'm hoping something-- where did we-- oh yeah, here. We show that A1 was similar to A0, and I'm hoping that's probably still true even-- the shift didn't mess that up. Let's just try. So A0-- that's R0, Q0 plus sI. And now what am I going to do? I'm going to-- what did I do before? I figured out what Q0 was from this. You remember? So this R0.

Now I have to put in Q0. But Q0 is this thing inverse times this. Is this going to work? I'm hoping, but I don't think I wanted to get that. No, it's not Q0 there. What do I put it here? And if it doesn't work, we'll leave it as an exercise.

AUDIENCE: [INAUDIBLE]

GILBERT STRANG: Yeah, because I didn't start right somehow. But let me just push along to see what happens. So I'm plugging in for Q0 here. I'm plugging in this matrix, so it's shouldn't have inverted it, times R0 inverse. Who knows? It might work. Who knows? So that's the R0, Q0. Right? Is everybody with me? Sorry about the-- R0 inverse. And then I have to add sI.

So what have I done? I've just pushed on, believing that this would work because it's the method that is constantly used. Now, do I have-- what do I have there? Is it working? This is R0 A. That was, of course, A0. R0, A0, R0 inverse. R0, A0, R0 inverse. Good. Minus s. What have I got there from the R0 minus sI R0? What is that?

AUDIENCE: Minus sI.

GILBERT STRANG: That's minus sI. Ha, look. Success. The R0 cancels the R0 inverse. So that term from that, that, that is minus sI cancels plus sI. I'm finished. And lo and behold, we have the same similarity. So we messed around by a multiple of the identity, and it didn't-- it actually makes the thing converge faster if we choose the shifts well. But basically, the same idea is still working.

So that's the QR method. Well, that's the method. We haven't shown and won't show that-- except for this half-completed example, I don't plan to prove that the lower triangular part begins to disappear, gets smaller and smaller and smaller, and then the eigenvalues pop up on the diagonal. It's amazing. Amazing.

Now, is there any other improvement we can make? So that's the method. And where is the work in using that method? Because that's what we always focus on. Where are we spending computer time? Well, we're spending computer time in doing the factorization. So it didn't cost anything to shift by the identity, but then we had to factor that into Q0, R0. Then it didn't cost much to multiply them in the opposite order. So the work was in QR.

So could we think of anything to improve that aspect? Can we think of anything there? And then we've got a really first class method. Well, the matrix A-- A0, the matrix we started with-- had some zeros that allowed us to skip steps in doing the QR factorization. So what am I going to say? I'm going to say if A or A0, our original matrix, has a bunch of zeros-- let's say it's got a whole lot of zeros there. Maybe it's--

Well, OK. I overdid it here. I know the eigenvalues right off. But so the truth is I can't-- saying that if that is not going to happen. But we can get zeros with one extra diagonal. That turns out-- so here is the main diagonal. Everybody's got his eye on the main diagonal. And one diagonal-- I can get a lot of zeros, but I can't by simple computations. And I'll show you how to get one.

But I can't get all those to be 0, because then I would have the eigenvalues right there. Well, how do I know that I can't? In elimination, ordinary solving AX equal B, you really do get to an upper triangular form. Some operator, some elimination steps you plug away, and your matrix becomes upper triangular U and you're golden. But that's too much to expect here.

In fact, we know we can't do it by simple steps because if we could do it, if we could get to a U with a whole lower triangular part 0, we would have found the eigenvalues. And we know that the eigenvalues solve a system-- solve an equation of nth degree. And we know-- somebody proved a century or more ago-- that you can't solve an nth degree equation by simple little steps. Do you know who that was and that you know that fact and what degree does it apply to? So that's an important fact that you pick up in math. Yeah?

AUDIENCE: The 5th [INAUDIBLE].

GILBERT STRANG: 5th degree, yeah. So 5 by 5 and up would be-- this is impossible. Impossible. There is no formula to find a simple formula for the lambdas. And similarly for the sigmas for singular values. So the eigenvalues is definitely a level of difficulty beyond AX equal B, the inverse matrix, or something, that pivots. All that you can do exactly if you're doing exact arithmetic.

We cannot find the lambdas exactly, but we can get as close as we like by continuing with the QR method. So yeah. In other words, we can't-- we have to settle for-- if we want to, like at the beginning, improve our matrix before we start doing that stuff, we can get it with one extra diagonal. And do you know what kind of a matrix, whose name-- I don't know why.

AUDIENCE: Upper Hessenberg.

GILBERT STRANG: Yeah. Say it again.

AUDIENCE: Upper Hessenberg.

GILBERT STRANG: Upper Hessenberg. So upper is just like upper triangular. It's up there. But key person's name is Hessenberg. As I say, that's a Hessenberg matrix. So Hessenberg matrix is a matrix with one triangular plus one more diagonal, but lots of zeros. Order of N squared. Something like almost like half N squared-- not quite, but close-- zeros.

And you could show those zeros stay zeros in QR. So that really pays off. It cuts the work down significantly. So that's the-- full QR method is Step 1-- reduce A to Hessenberg form with these zeros. And when I say reduce, I mean find a similarity transformation, of course, because I want the eigenvalues of this to end up the same as the Hessen. I want to keep the same eigenvalues as I go.

And then, Step 2 is QR on this Hessenberg matrix with shifts. So that's the code that would be programmed in eig(A). That's what Matlab and-- well really, Matlab is appealing-- like other matrix systems-- is appealing to LAPACK and LINPACK. A team of professional numerical analysts really spent a lot of effort and time. The book *LAPACK* has 10 authors, and you can download any of these codes, like the eigenvalue code. So that's where Matlab, naturally-- that's the Bible for code in linear algebra. I think it's interesting to know.

And there's one more good thing to tell you about this method. And it applies if the matrix is symmetric. If the matrix is symmetric, then if we check all this, we could find that the matrices stayed symmetric. If A0 is symmetric, I can check-- you could easily check through this and you would discover that A1 is also symmetric.

It turns out you could rewrite this with a Q0 in a Q0 inverse on the other side, but that Q0 inverse is the same as Q0 transposed because it's an orthogonal matrix, and symmetry would fall out. So if it's symmetric, and it's in his Hessenberg form and it stays symmetric at every step, what can you tell me about a symmetric Heisenberg matrix?

AUDIENCE: [INAUDIBLE]

GILBERT STRANG: It's only got-- yeah. You just erase all these. If they are zeros there and if the matrix is symmetric, then we can safely predict that it will only have one diagonal above, one non-zero diagonal above the main diagonal. In fact, it will say symmetric. So now I should write "symmetric Hessenberg matrix," and equals tridiagonal matrix. Three diagonals.

So now you really have reduced the time to do QR because you've got a tridiagonal matrix. It'll stay tridiagonal in all these steps. So you're working with just three N numbers. Well actually, two N because the diagonal above and the diagonal below are the same. You're working with just two N numbers instead of order N squared, and it just goes like a bomb. So that's eig for symmetric matrices. And you see that it was all based-- that really the heart of the algorithm was QR. So that's my-- that took half the class to report on the favorite way, the eig way, to find eigenvalues.

Oh, I should say something about singular values. So singular values. Of course, the singular values of the matrix are the eigenvalues of A transpose A-- square root of those eigenvalues. But you wouldn't do it that way. You would never form A transpose A. Oh, I didn't mention the other thing you would never, ever, ever do, so let me just put it here like in disgrace. To solve that equation is like, OK, back to first grade, because that's not-- that's very bad.

A determinant-- first of all, it's extremely slow-- extremely slow. And the determinant is packing all this N squared pieces of information into N coefficient, and it's hopelessly ill conditioned. Yeah. You lose information all the time. So really, if this is going on camera, it better go on camera with an x, because you don't do it.

Yeah. So where was I? Singular values. So A transpose A. So again, let's think about what you could do at the beginning before starting on QR for A transpose A or for the matrix A. What could you do with orthogonal matrices? So I guess-- what did we say about symmetric matrices?

So here's what I said about symmetric matrices. If you'll give me a symmetric matrix, I can in just a simple number of simple steps make it tridiagonal. I can't make it diagonal because then I'd be finding the eigenvalues and Abel, who was the first person to see that that was impossible, forbids it.

So let me let me write down what I'm saying here. If I have a symmetric matrix S, I can find a bunch of Qs and Q transposes, and I can put them all together into one big Q, and it's transposed. And what do I know about the eigenvalues of that matrix? Q is orthogonal always.

So what can you tell me about the-- this is the same as QSQ inverse, and therefore the eigenvalues are the same. It's similar to S. And it becomes tridiagonal After? I find a good Q. Therefore, same lambdas. It's tridiagonal with the same lambdas.

Now, what am I thinking about here? I'm thinking about-- tell me the corresponding possibility about singular values. I wanted to do something to my matrix. Now, I'm always taking a general matrix A. And I'm looking for its singular values. And I'm looking to simplify it. And what am I allowed to do? Yeah, I guess my question is-- similarity transformations left the eigenvalues alone.

What can I do that leaves the singular values alone? That's a fundamental question because it was so fundamental for eigenvalues. By doing this, a matrix and its inverse, I got something similar, and I checked even in this class that the eigenvalues, same lambdas. Now I want a whole line that ends up with the same sigmas. And I want you to tell me what I'm allowed to do to the matrix without changing the sigmas.

So this is a-- maybe don't shout it out immediately. Let everybody think. What am I allowed to do to a matrix? Every matrix has got these singular values, and now I want to make it a better matrix with more zeros or something. If I do that to it, does that change the sigmas? Can I do more than that to it? What can I do? What group of matrices will have the same sigmas as my starting matrix A? So that's a basic, basic question about singular values and the SVD.

So let's think of the answer together. So it's connected to the SVD, so let me remember the

SVD. The SVD-- I have some orthogonal matrix. Then the singular value of the matrix-- SV for singular values. And then another orthogonal matrix. What could I do to that equation that would not touch this guy?

So I'm asking, what invariants? Because not touching it means leaving it not varying. So I'm looking for under what operations are the singular values invariant? When I was looking at eigenvalues, this was the operation. Well, it didn't have to be orthogonal-- something, then its inverse.

But now, what is it up there? What could I do to that matrix A? Could I multiply it by Q? Could I throw in a Q maybe not even on the other end? If I throw in an orthogonal Q, do I change the singular values or do I not change them? Fundamental question. The answer is no, I don't change them. I'm allowed to do. That because here's an orthogonal matrix, a Q times U. If both of those are orthogonal, then the product is.

Everybody knows that a product of two orthogonal matrix is still orthogonal. Better know that. Better know that. So if I have an orthogonal matrix Q and an orthogonal matrix U, I claim that this is still orthogonal. And how do I check it? Well, I use some test for orthogonality. What would be the-- what test do you like to use? The inverse is the same as that transpose, do you like that test?

So I'll invert it. QU inverse. Of course, for any matrix that's U inverse, Q inverse. But these were separately orthogonal, so that's U transpose Q transpose. And that is the same as QU transpose. So I used the orthogonality of U and the orthogonality of Q to conclude that the inverse is the transpose. So the answer is yes, I could do that.

Now, with singular value-- with eigenvalues, I had to multiply on the other side by Q inverse or Q transpose. Do I have to do that now? No. What can I do on the right-hand side? I can multiply by-- I can leave it alone. Then it has the same singular values because it's the same sigma in there. If I have a orthogonal matrix times a diagonal times an orthogonal, that diagonal is-- positive diagonal is going to be sigma.

So what can I do on this side? I can multiply by any orthogonal matrix on that side too. So let's call this guy Q1 and this guy Q2. I still have an orthogonal matrix there, orthogonal matrix there, and the same sigma popped in the middle. So that's what you're allowed to do. That gives us more freedom. Before we got-- when we had to do similarity transformations with the same guy, we got it to be tridiagonal.

But now, we're allowed to do more stuff. We're allowed to use different orthogonal matrices on the left and right. And we can reduce it even further from tridiagonal to bidiagonal. So the first step is getting zero. The step of getting zeros reduces it all the way to that, with all zeros there. So it's easier. Then I work on this. This is the matrix I work on using a QR type idea, some method like that.

So everybody's seeing that our algorithm has got two stages. One is get a lot of zeros and get them in places that will stay zero as Part 2 of the algorithm gets going. And then, run Part 2 of the algorithm. You staying with-- each step is very fast now because doing a QR factorization is fast. Was there a question? Yeah. So I would call this bidiagonal, of course.

And everybody recognizes that if I have a bidiagonal matrix-- call it A or A0 or whatever-- then what do you think about A transpose A? What would A transpose A-- if that was A, what could you tell me about A transpose A? Could you multiply matrices knowing where the non-zeros are in your head and get an idea of where-- and so if I have a bidiagonal matrix A, then implicitly in the SVD, I'm looking at A transpose A. And what would be true about A transpose A? It would be tridiagonal.

So what I've done here and what I've done there just match up. You can operate-- if you don't want to change singular values, you can get all the way to here. But then, to find those singular values, that would involve A transpose A. It would be symmetric and tridiagonal, and then you'd be in that game. So those are the basic facts of eig and SVD for matrices of order up to 1,000, say. I'm not enough of an expert to know where--

Maybe higher, because in perfect math, it's going to take infinitely many steps or Abel would be very surprised. He would see you solving for eigenvalues an nth degree equation by a whole lot of little steps and getting them exactly right. That won't happen. But you get them within epsilon in a number of steps that's like N cubed. So that's pretty impressive. The eigenvalue problem is being "solved," in quotes, by a fast method that gets you a good answer within a tolerance in N cube steps.

So that's great as long as N isn't too big. And then, when N is too big-- which, of course, happens-- you have to think again. So this method is a giant success up to large matrices, but then you have to think again. And what is involved in thinking again? Well, I guess more thinking. So what do you do if the matrix is bigger?

I guess that Krylov would say, use my method. So Krylov would say-- especially if your matrix was sparse. Can we just remember what Krylov was? Krylov started with a vector b, multiplied it by A, multiplied that by A, and got up to, let's say, A to the 999b. So now he's got-- Krylov has got 1,000-dimensional space. He's got a basis for it-- 1,000 vectors that span 1,000-dimensional space.

And he'll look at the matrix A only on that space. In other words-- I won't go into detail about that. He restricts the matrix to this 1,000-dimensional space, and he hopes that it's captured. We hope that the eigenvector is almost-- is virtually in that space. And actually, I wouldn't go to-- let me take a 9 out of that. 100-dimensional would probably catch the eigenvector.

So if the eigenvector is virtually in this space, then we can look at a matrix of order 100. We can bring A down to just see its action on that space. And any piece of-- so here is a-- so I look at vectors v, which are some combination-- c1 b plus c2 Ab plus c3 A squared b, and c100 A to the 99th b. Plus an error. And I'm going to ignore that error because I've gone up to dimension 100. I'd probably say it's pretty safe to ignore that error.

And then, in this space, just looking at the matrix A-- so wherever A to the 100th comes in, forget it. Just think about the matrix A as multiplying vectors of this kind in this space. Then I have 100 by 100 eigenvalue problem. And so the big matrix A is reduced to a matrix of size 100 by-- do you see what I'm saying even? So I'm not giving the details.

Think of a matrix A of size a million. And you apply it to Krylov vectors-- so I call them little k for a Krylov vector-- in this 100-dimensional space. So they have a million minus 100 0-components, you could say, this k. This is in the Krylov space. This is A, a million-- k, 100. It's a full-- so it's got a million components, but it's out of just 100-dimensional space. So when I multiply by A, it'll be mostly in-- partly in the Krylov space-- k100-- and a piece out of k100.

And I just ignore that part of the matrix. So I have 100 by 100 problem, and I solve to find the eigenvalues. And they're a pretty good approximation to the eigenvalues, to the, hopefully, like the lowest 100 eigenvalues. I'd like to know that, but I might not be sure that this idea would give me the lowest 100-- the first 100 eigenvalues of the million, of the matrix of size a million. I'm just taking a few minutes here to wave hands about what Krylov idea would do. And I probably won't mention Krylov again in this semester.

So what it can do is look at this particular type of space because we can get a basis for it quickly. Just multiply again and again by A. Then we can orthogonalize that basis. That's

Gram-Schmidt in some form. We're always going back to Gram-Schmidt. Then I have 100 by 100-- I have a subspace of size 100. I look at what the matrix does in that space, and that I could look for-- I could find eigenvalues restricted to that space. They wouldn't be the perfect eigenvalues, but they would be accurate.

So I didn't know it would take one class time to talk about finding eigenvalues and singular values, but we did some important things. We remembered that similarity is the thing to check, thing to preserve, because it doesn't change the eigenvalues. And then we-- for singular values, what was the thing? You could multiply left and right by different orthogonal matrices. And somehow, maybe that doesn't have an established name, multiplying left and right by a Q1 and a Q2 transpose. But the idea is clear, and that doesn't change the singular values.

We're ready to move now into-- maybe our next step, which we don't spend a long time on, will be random sampling. What if your matrix is just way too big? So that's a very new idea, very different idea in numerical linear algebra, is just to sample the matrix. Could you believe that the answer is going to come out right just for a random sample? Well, the odds are in your favor. So that will be Wednesday, and then we have lots of new-- we'll move onward after that. See you Wednesday. Thanks.