INTRODUCTION:

PROFESSOR: Specific problem, and it's linear least squares problem, but it's got two terms. So we're used to minimizing Au minus b square. That gives us the least squares solution u hat to linear system. And usually the reason we have to go to the least square thing is that there's no exact solution. Probably A has more equations than unknowns. A is long and thin, and there's no exact solution, so we look for the best solution, and we call it u hat. OK. But there are a lot of problems in which a second square appears. There's also a Bu equal d hiding in the background. And so we really have like two sets of equations. And we multiply that second square by some factor alpha and that wise choice of alpha is usually a big part of the problem. And I want to speak about some of the applications of this area. So from that point of view of the normal equations, the system that you actually solve, you could say no problem. If we knew how to do this, then we certainly can do both of them together because instead of A transpose a showing up, we'll now have A transpose a plus alpha B transpose b. That'll be the positive definite coefficient matrix on the left side. And then on the right side instead of just the usual A transpose b, this term is also going to give us an alpha B transpose d. All I'm saying is we don't need any new mathematics to reach this normal equation. With the sort of the two terms normal equation.

And another way to think of exactly the same thing is we're looking at the least squares problem, where the two matrices A and B are multiplying u. And we have to bits of data being B and d, and all were doing is the usual thing but with a weight in here. And the weight is the identity matrix for the A part, and it's alpha times the identity matrix for the B part. So this is our C right. This is our C, just to say that really the notation that the formulation we have allows us to take this step, so C appears here and A transpose was CB, c appears over here too, just as always. OK.

But there are important questions. And of course always the first important question in applied math is what problem are you solving? Why have we produced this class of problems? And I have two answer. Let me just first, so we are sure what the shape of these matrices is. A as always has more rows than columns. Of course u is n by 1. it's just a column right. But A has too many equations, too many rows, for us to get an exact solution; v on the other hand has few rows. It might even only have one more. It's very common to add on one constraint or one term in regularizing the situation. Anyway p is relatively small. So the total matrix AB has m plus p rows, and the same n columns, and we're ready to go. But the two parts are different somehow. They come for different reasons.

And now I wrote down here two places they come from. And these are big applications of applied math. And one of them produces small coefficients alpha. And what's the purpose of the Bu minus d term in that case, with just a small alpha? The problem is that the A transpose A part is nearly singular or is singular. So that the usual normal equation without the B would be in trouble, and this of course happens pretty often. So to the idea of regularization is get some control of the solution by putting in another term that keeps some control over u, and stops it from just taking off as what happened where the or original normal equations would have a very large u hat. So we're just like adding a little steady part that keeps it a bit under control. And so the A transpose a is nearly singular in ill posed problems. It's like giving aspirin to an ill posed problem, right. You don't fix it, but it can operate. OK.

And where do ill posed problems come from? And I just wanted to say that I think that fundamental ill posed problems in science is given positions. Suppose we know that a mass let's say, is in certain positions at certain times, find the velocity. So we often in applications have some way to know position, and want to know velocity. And maybe you realize that that problem is not well posed, because velocity take the derivatives. And if you take the derivative, that's not a good operator to invert. Taking the derivative makes things very rough. All sorts of cases we're looking for the velocity, and we only have positions. One that I think about is from GPS. So

GPS uses space based satellites, as you all know, to give you very accurate positions. And somehow out of those positions, you get pretty accurate but not of course as accurate as the positions, but you get decent velocities. And how? And there is an example where you want to know the motion of course, to ask for the acceleration would be asking for yet another derivative. You see why the derivative is an ill posed thing?

Let me just say ahead of time, I'm going to make today's lecture about direction number two, not the ill posed problems. So I'm just like throwing in some comments about the ill posed problem, and then I'll have a weekend to think about those. And then next week I'll come back to this ill posed problems. And specifically they often come from inverse problems, is a big source of ill posed problems that need regularization. It's just a very large class of equations. I mean I was just going to say about the derivative example. Why is that so unstable? Well from the point of your finite differences, if we have positions how do you estimate velocities? You take a difference quotient, right. You take the position at this time, the position at a close by time, and you divide by delta t. That's a reasonable start. But dividing by delta t, that small number, is producing big numbers. Any errors in the position are multiplied by 1 over delta t and blown up. And similarly in frequency space, where the functions that we think about are the functions like either DI KT the derivatives brings down the factor K. So high oscillations, that's the point. Oscillatory functions can be pretty small, but their derivative can be enormous. So it's that oscillation which is often associated with noise in the measurements. You know, noisy measurements are jumpy, and when we go to take their derivative or their finite difference, we get big answers. Anyway for me that's the model ill posed problem to find velocities.

And how to do it? I mean a lot of thought is going into that. Let me leave it there, and come back to it. But I say all this just to emphasize its importance. Not that we'll completely solve it actually for GPS or for any other thing, it's just all we can do is medicate. OK. Now this is the one that we can really solve. So this is a different application entirely. In this application, this second term Bu equal d, is something important, something that we want to enforce. It's a constraint you could say. And one way to enforce it which fits this pattern is to take alpha very large, right. When we take alpha large, we're putting a really heavy weight on that Bu minus d square, and when we minimize, that weight will forced Bu to be pretty close to d. But of course Bu equal d doesn't determine u. Everybody's got that picture clear? From BU equal d has many solutions. And so the real problem that we're trying to solve is enforce Bu equal d. But among those solutions, pick the one that minimizes the first square, Au minus B square. So you see the difference? You're trying to enforce something that the physics or the geometry, or whatever sources, has to be through. And you can do it. And you're left with lots of options. And then the combine problem attempts to pick the right u. OK. So that's the application number two that I want to speak about today. And actually I want to give several ways to do it. It's a very important problem.

And one way will be to actually solve Bu equal d. Find those solution. And you may say well that's what we learned in linear algebra, that's the very foundation of linear algebra is there a particular solution, right. Every solution is of this form particular plus null space. Maybe I'll just point to the start of that approach. So want to solve Bu equal d. And I'll come back to this method after dealing with the least squares I approached. But here's really the direct approach. That if I solved Bu equal d, then there's a particular solution that solves it. And then you can always add on the general solution which is, sorry add on the null space solution - the solution of Bu equals 0. And Bu equals 0 has lots of solutions. So we would have to find them. OK. I mean that's what 1806 would naturally do, but actually never, I'm ashamed to say, but I didn't do it in 1806. I never actually said how I would scientifically compute in a stable way the solutions. OK. So I think that will be important. But that's not the only way to do it. That's called the null space method. And sometimes it's the right choice, sometimes not.

This would be called the heavy weight method, right. Put on a very heavy weight and solve a standard problem. OK. So let me follow that one up. And then they'll be a third method. And maybe there's going to be space on the middle black word for it. And what would the third method be? That will be use Lagrange multiplier. This thing is a constraint. I'll enforce it by a Lagrange multiplier. OK. That's coming next. The way I'm enforcing it right now is by a heavy weight. OK. One reason for the popularity of this method is you don't have to do any new thinking. You just create these equations and solve them. Where the other methods maybe ask us to think separately about the constraint. Here we don't have to things separately, we just create this normal equation, we solve it, and we get an answer u hat. Maybe I should call it u hat alpha, because it depends on the eight alpha certainly, which we hope is near the exact solution. The exact solution being the one that exactly solves Bu equal d. Because u hat alpha will not exactly solve the Bu equal d. But we can find solutions that do, and then among those we can minimize Au minus b square. OK. So just a word about this heavy weight method. OK.

Well first an interesting point. A point that I think it's sort of interesting. I want to let alpha go to infinity and see what happens, right. Everybody figures that as alpha goes to insanity, I'm going to get the right answer. Because as alpha goes to infinity, it's going to more and more enforce the constraint Bu equal d. And then with that constrain enforced the other part of it will find the best u and that's great. But let alpha go to infinity in this equation, and what happens? So this is just like a side comment just to say alpha, you know, taking a limit you got to think about doing it right. Well let's see, if I let alpha go to infinity as it is, that'll be infinite that infinite, I won't know what's going on. Let me divide by alpha before I let alpha go to infinity. So if I just divide everything by alpha -- can I do that with an eraser here? I'll divide by alpha. So there's a 1 over alpha here. I divide this by alpha. And this has a one over alpha there. And now if I let alpha go to infinity, I get something sensible. This goes to 0, right, alpha going to infinity getting bigger and bigger. This goes to 0. So what do I get in the limit? I get that this equals this in the limit. So shall I put that up here? Well I'll put it here, because I don't like it frankly.

So I'll just squeeze it in this little spot. That if I let alpha go to into infinity, so 1 over alpha goes to 0. I get B transpose B, u hat infinity shall I call it? Equals b transpose d. And I guess what I want to say is from that I don't learn a whole lot. because B transpose B is a singular matrix, B transpose B is a matrix of only rank p, it's very singular right. B had this crazy shape, long and thin. B transpose B will be tall, B transpose B will be a large matrix, but its rank will only be p. It's an n by n matrix of rank p, and it's singular and who knows what's going on there. That little side issue was simply to say that you can't just let alpha go to infinity and central equation there, and expect to see what's happened. OK. So somehow there's more to it than that. So let me put alpha back where it belongs, and think again. OK. And I guess by thinking again, I might as well think in terms of this way of writing it. Because I recognized this right, this is exactly the framework that we've developed.

So this is the least squares problem. I just want to write down the saddle point matrix that goes with this least squares problem. What is the saddle point matrix? Do you remember? The saddle point matrix S. So now I've got an A and a B here. So it's going to be larger than I have my usual 0 block. And I have my usual A transpose B transpose block. And what block goes there? That's the C inverse right. It's our usual C inverse A, A transpose 0 that we're totally accustomed to. But now A has grown into AB, 0 is still 0, a transpose is still the transpose, and up here is transpose inverse, and since C was this, C inverse will be the identity, and the identity over alpha. OK. So that's my S alpha you could say. So my equation is alpha, what's written as a block equation. What are the pieces of it? u is the guy that I'm looking for, the u hat alpha. And there was a Lagrange multiplier that came. You remember that's how we got to a block form from a scalar form. And I guess I usually call it W, so I'll stay with W for here. OK. So that's what multiplies Wu. And I think it gives a B and it gives a d from this Au and Bu, and I think here if gives a 0, because we didn't have any. OK.

What am I doing here? I'm just writing the problem in a way where I can let alpha go to 0, and see the limit. So let alpha go to infinity, this is transpose part. So this will go to 0. So this approaches the S infinity, W infinity we could call, u hat infinity is now, well you see what the limit, is that's 0 in that block. This is A, this is B, this is A transpose, this is B transpose, this is our usual 0 block, multiplying our same W infinity, u hat infinity, equaling our same B, d, and 0. This is the limiting equation. And it's great. This is the equation determines the limit, as alpha goes to infinity that it determines the best u. This is the problem that we really want to solve. Maybe that's what I should say. Do you see the constraint Bu equal d in here from this middle block row? That say 0, 0, Bu hat is d. So we've introduced the constraint. The first part is W infinity with an Au infinity, that's the usual error term, the thing that we probably can't make 0. And then this is the usual Legrange multiplier terms from there.

So I've spoken pretty quickly here, and let me just conclude. This is the limit equation, is the correct limit equation. This is the limit equation that we want to solve one way or another. And taking alpha large is one way to get near the answer, but we'll look at other ways now. So this is really the correct equations to solve. The saddle point Lagrange multiplier route. OK. So let me summarize what I've done so far. My problem is when Bu equal d is a constraint that I would like to satisfy, and one way to do it is to take alpha, you know, pretty near the largest number that the machine will hold, say 10 to the 15. Put a really heavy weight on this. But of course when you let alpha be 10 to the 15, you can see that there's like some possible problems here. When you let alpha have an enormous weight, you're really tilting this matrix so strongly, you know, you couldn't let it be 10 to the 20 in single precision or you'd wipe out A transpose A. So it's a balance here. So I guess probably a lot of a numerical analysts would say wrong way to do it, the right way is solve this equation or, else do it this other way. But a lot of people with codes say OK, you know, you're going to be a nervous nelly I'm just going to use my code. And that's quite normal, quite human response. OK. And this will frequently succeed. OK. So that's one method to do it, not the method that professionals in numerical analysis -- maybe I'm thinking for example the book by Golub/van Loan, if you know that book, that would discuss this problem. And it would actually discuss this third method, this null space method of solving. OK.

Maybe I'll go to that null space method. So this was one way. Another way is solve Bu equal d. And remember again, we only have p equations we have n unknowns, so there's going to be freedom in the solution. So we have to identify a particular solution, there's a lot of freedom in that particular solution, and then we can add to. The null space is going to be n minus p dimensions, n minus p degrees of freedom in the null space. That's the dimension of the null space, n minus p. I'm assuming that b has full rank p, but p is a small number compared to them. OK. So how do you find a particular solution? How do you find the null space solution? As I said, that's what I should be explaining in 1806. And of course, we do it in 1806, but we do it with a 3 by 3 matrix, and we practically, you know, we do it by hand, where here we're talking about matrices of order thousands or millions, we don't do those by hand. And we better not do it in an unstable way.

So the question is what's a good way to do it? And really the heart of modern numerical analysis is orthogonalize stuff, get orthogonal vectors. Because if you have orthogonal vectors, they don't get out of scale they. The numbers involved don't become unstable. And the standard orthogonalization process is Graham Schmidt, that's right, those are the words we all think of. If I have a bunch of vectors, I have to make them orthogonal, I want to make them orthogonal, well Graham Schmidt is what we think of. But actually math lab doesn't use Graham Schmidt, doesn't use the usual Grant Schmidt as Graham Schmidt thought of it. Mat lab goes a different route to the same conclusion.

So let me just remind you what Graham Schmidt produced. And let me put in the name of the numeric analyst long after Graham Schmidt, it's Householder, you know, the guy from Tennessee with good ideas. So he had another way to the same answer, which is this factorization. So we take our matrix, often it's A in 1806, and we factor it, we want to or orthogonalize its columns. So the columns of A get orthogonalize into the columns of Q. So this has the orthogonal columns. And then of course there's some connection between the original columns and the orthogonal columns, and that connection is by triangular matrix R, upper triangular. I don't know if you remember that from 1806. What I typically do is I explain Graham Schmidt as they knew it, and then at the last minute I pull Q and R out as a way to express the result. OK. So it's the result we want, and not the particular Graham Schmidt way to get there, and Householder produces a better way to get there. OK.

But the main point is that if a matrix has independent columns, or even if it hasn't, but if it has independent columns and we know everything about it, we can orthogonalize those columns. Here's what I'm leading to, this B transpose I'm remembering, has this shape because B had that shape, so it's B transpose that I'm going to do Graham Schmidt, Householder use. The command in math lab is QR equals, with Graham Schmidt we could have used the letters G and S but since we don't use their actual anymore, we could use the letter HH for Householder or something. But it's QR of, in this case, B transpose is what we want. OK. Very frequently used command in math lab produces is Q and R. And it produces a square matrix Q, where these columns, the columns of the first part Q1 transpose are orthogonalize versions of these columns. And the R just tells us the connection between them. Then it also produces, and this is handy as you'll see, the algorithm also produces n minus p more columns, that are orthogonal to these guys. So it produces a complete orthonormal basis, a complete set of n columns altogether. Q1 transpose has the column that really are associated with these problems. And these are going to be associated with the null space. So out of this I see that actually B transpose is Q1 transpose R. So you can say this is the reduced factorization with only p columns, and this is the full picture with the other n minus p columns that are orthogonal. And the reason that's handy is they tell us about the null space.

So now I want to identify out of this a particular solution and the general null space solution. OK. So what are those? So particular solution is going to use this part. So let's see, I want a particular solution. So B transposing that is R transpose Q1. OK. So now I'm prepared to solve. Step one is the particular solution. I want to get be Bu particular equal d. OK. But now I know B is nicely factor. So this is R transpose Q1 u particular equal d. So now comes the computation the code has to do. It has to invert that to get Q1 times u particular equals R inverse transpose d. So it had to solve a triangular system, but of course a triangular system is quick to solve. That's a good part here. And then the final step to get u particular, I have to put the inverse of that guy over there, because this has orthogonal column, that's just Q1 transpose. So there we go. That's the inverse of R. So that's what I should've done in 1806 and never did, and you get to see. What's a convenient particular solution? Everybody knows we got a whole selection of particular solutions. We want to choose one that's nice and stable. And the reason it's stable is that it works with orthogonal columns, orthonormal even, and triangular matrix for which linear systems are highly active. OK. So that's the particular solution.

Now what's the null space solution? What are the general solutions to null space part? What are the solution to those? Well I can just go down the same steps. This is R transpose Q1, u null space equals 0. I multiplied both sides, this is a nice square and veritable matrix, and multiplied by its inverse kills that. So now I have Q1. Q1 is really the heart of B. So what vectors are perpendicular to Q1?

I hope I've got this right. It's easy to mix up a transpose in the process. So let me just pause to be sure I'm doing it correctly. OK. I hope. Did I check that I get it right? Yes. OK.

I could have written of what B is here, since I have B transpose as a product. B is R0, Q1, Q2. OK. And I want to multiply by u null and get 0. OK. So what should u null be? This part is giving us a 0, so this is like gone. So you see the two are the same. So what vectors are perpendicular to those? The answer is the u null is a combination of the columns of Q2 transpose. It's the Q2 part that's telling us about the null space. It was the Q1 part that gave the particular solution, it's the Q2 part that gives the general solution. In other words, u null is Q2 transpose times any vector, let me call z, this is any z. OK. Now this has my n minus p degrees of freedom. Sorry I'm trying to do quite a bit here. I'm trying to say how you actually solve rectangular systems when they're not determinate. There are many solutions. This is a good particular solution to find, and this is a good way to find the general solution, the null space solution. This is a combination of the other columns. OK.

All right now we're done really, because I now know what u looks like; u looks like this part which I've computed and this part which has the freedom. Let me put those two parts together. So now I want to minimize -- so I'm near the end here -- Au minus B, but Au is u particular, and I have u particular here. Q1 transpose r minus transpose d, that's u particular, plus u null, and that's this. This u null was also here; u null was any Q2 transpose z, right. All that is u, AU minus B. OK. Up to possibly screwing up on some transposes, this is the right method. So this is a fix solution. I just want to write that as a different way. Minimize A Q transpose z. It's now we're minimizing over the z's. So u had n components, but somehow p degrees of freedom were used up by the constraint Bu equal d. And we have the n minus p true degrees of freedom are in the z. So there's this minus the B. This is all known stuff, A Q1 transpose R minus transpose d square. OK. I'm there.

So this is a standard minimization problem. Minimize, shall I call this A tilde z? And I'll call all this stuff B tilde. And the solution is found from the normal equations A tilde transpose, A tilde times the best z, I'll put a hat on it to emphasize that it's the great one, is A tilde transpose B tilde. OK. Finish that process without leaving myself a lot of time for the other method.

Conclusion here that after you've done the QR stepped, the QR command, and then after you solved a linear system with the r transpose, and you've multiplied by Q's, and you've ended up with this problem with a new matrix A tilde and B tilde, then you just do the normal equation. The web will have the code that takes those steps, reaches this conclusion, and solves it. OK. So that's the null space method. So z has p minus n components, n minus p components if p is near n, then they're not many z's and this is highly efficient. OK. So the null space method is one way to go.

Can I just in the remaining minutes go back to the Lagrange multiplier idea? so what's the Lagrange multiplier idea? So let me write the problem again as Lagrange would like it. Minimize Au minus B square subject to Bu equal d. That's the problem we're solving. I should of written it earlier. Let me put a star here, because this is our problem. OK. So one way to tackle it was take that constraint give it a heavy weight. That was method one.

Method two was solve this constraint in full detail, get the z's that remain as degrees of freedom, plug in u particular plus u null space into here, and then you have a problem in the z. That's method two. Now, so method three is Lagrange. So method three would say OK, what does Lagrange do? L, we call it the Lagrangian, e takes this Au minus b square, and adds to it some Lagrange multiplier, and all the u's are maybe the standard lambda, times Bu minus d, right. That's Lagrange's idea. You recognize Lagrange's idea. Takes the constraint, multiply it by a multiplier. In fact this is p constraints, so p lambdas. Lambda's a vector of p multiplied. Not just a single one, because we've got the p constraints. And now what does Lagrange do? He sets the derivative dL d lambda. Well, so let me do the dL du first. He sets dL du to 0, and dL d lamdba to 0. I could've started out with this method, because it's going to lead us to the equations faster.

What equations do we get from dL du equals 0? What's the gradient with respect to u? That gives us A transpose Au. Oh, probably we want a 1/2 here, so that numbers come out right. We get A transpose Au, and another u part will be the B lambda. Taking the derivative of u will produce a B transpose lambda out of that. Yeah a B transpose lambda out of that. And then in here will be a linear term in u that we might as well put on the right hand side as a transpose B. Familiar. OK.

And what about dL d lambda? Well that's just our constraint, Bu equals d right. Having built in the constraints, when I take the derivative with respect to lambda, the constraint just comes back again. So this is now method three. Solve that system?

And I guess what I want say in the remaining 30 seconds is that solving this system is the same as this one. Those two are exactly the same. So that's a system with three parts, But maybe I can even get there. Can you see that if I take this part and I said subtract A transpose those times the top row from the bottom row, what will that give me? Let me just hope that it works, well I won't actually. Time is up, it's asking too much to do even this one piece of linear algebra tat can be in the notes. So this system that we got as the correct limit equation is exactly the same one that Lagrange gets. So that's one way. This is a system with n plus p unknowns. That's the price you pay for going Lagrange's route. You had add p unknown. This was a system with n minus p unknowns. That's because you're using the constraints to reduce the problem. And the original method one was a method with n unknowns, the unknowns and u. So you have the choice n plus p that Lagrange would like, and n minus p that Golub/van Loan would prefer. And usually it's method two or method three is recommended, but method one often used.

OK. So that's the lecture on this point. That's today. And then next week comes the whole class of problems like finding velocities from displacements, where alpha is a small parameter. And then after that come discussions of the completed project ones, and the upcoming extensions into project two. OK.

See you next week, thanks. Good.