

The following content is provided by MIT OpenCourseWare under a Creative Commons license. Additional information about our license and MIT OpenCourseWare in general is available at ocw.mit.edu.

PROFESSOR: OK. Good. So I decided to make today's lecture the one on linear programming and duality, which I'd planned for Friday, and give myself two more days to learn about ill-posed and inverse problems, and then come back to that Friday, so that we've studied the limits in those problems of α going to infinity or 0, but the scientific question when there's noise in the system is finite α and I want to learn more about applications and examples.

Can I also say I'm very happy to have had volunteers for Monday and Wednesday of next week to present, and if a couple of people might maybe volunteer for Friday, to share Friday, I'll be very grateful. So you could see me after class, put a hand up now, send me an email -- all those would be very good And again I would be thinking -- since it's just next week I'm talking about -- that it would be essentially a report on your Project One that you would use the overhead projector maybe, if that's preferable.

OK. So I think you'll like this topic. It's kind of specific but widely used -- linear programming -- used in business to maximize profits, to minimize costs. And linear means that the cost function is linear. That's an inner product -- $c \cdot x$. c is a row vector, x is a column vector, so that I'm following the conventions of this subject here to take these different shapes, so let me indicate what the shapes are. But the inputs are -- the data of the problem are -- c and A , an m by n matrix and b . So A is m by n , b is m by 1, -- right hand side -- and c is 1 by n . And then the unknown -- this is the thing that we're to find -- it's a column vector -- n by 1. OK.

And the point is there are constraints and those are linear too. So it's rather unusual to have a linear cost function. Right? Because when you maximize or minimize some linear function, well, the thing is just going up or it's going down -- or in higher dimensions the same -- and if it's going down, then the minimum is going to be at

the right hand end. Or if it's going up, the minimum will be at the left hand end. And if I'm in more variables, this idea will still be true that the minimum or maximum will happen at the edges, at the ends of the allowed region and this allowed region, called the feasible set -- so let me give name to this -- these are the allowed [? axes. ?] These are the constraints, and that set is called the feasible set, feasible meaning doable. So those constraints include inequality because we want finite intervals, finite regions in n dimensions. And I drew a sort of quick picture so that you have a model of this. So this is a picture with n equals 3 -- 3 dimensions, and so the constraints x greater or equal 0, x sub 1 greater or equal 0, x sub 2 greater or equal 0, x sub 3 greater or equal 0 -- that's what x greater or equal 0 means. It means all components. So we're in the quadrant right? We're in a quarter -- 1/8 sorry -- we're in an octant -- 1/8 of three dimensional space, the positive octant. And then if I draw maybe just one, just put in one equation, one plane, would cut off a piece of that octant, so that Ax greater or equal b , depending on the signs, but the feasible set could well be the tetrahedron, the little piece of the octant that's cut out by this plane. Or if our constraint was an equality, the feasible set would be the triangle. So Ax equal to b would lead to the triangle, and Ax greater or equal b would be, if we pick the signs correctly, would be the pyramid, would include also this corner, because there'd be some volume.

OK. So the feasible set is a polyhedron. It's like a polygon only up into n dimensions, so we use the word polyhedron. And it's got corners, and the whole point of the linear cost, the linear objective function cx , so this is just c sub 1 x sub 1 plus c sub n x sub n . That's what that means. If I take derivative I get constants. I don't set derivatives to 0 in this type of problem. I look at the end points, at the corners. And that's where the minimum and maximum will occur. So it's just a question of finding the right corner. That's the problem: how to find the winning corner.

It's an interesting competition between two quite different approaches: the famous approach -- so let me write these two. The simplex methods is the best established, best known, approach for solving these problems. What's the idea of the simplex method? The simplex method finds a corner. A corner is a case where we have some equality signs. A corner is the edge, the limit where maybe this one still has x

sub 1 positive but it's down in this plane so it has maybe x_3 as 0 for this guy. So that corner has $x_3 = 0$ and it also lies right on the plane so it has Ax equal to b . This corner -- well, I guess that corner has all these guys equal 0: $x_1 = 0$, $x_2 = 0$, $x_3 = 0$, but Ax inequalities over here for this corner that's hiding behind the face.

Anyway, corners are points where some of the constraints are tight or active and others are not. Well, you might say just check them all, but the trouble is there are lots of corners. If we're in n dimensions and we have m constraint equations, then the number of corners goes up exponentially. So there's no way to check all of them. So the simplex method had a better idea. The simplex method found one of them -- and already that's a little bit of a job to find a corner, but finds one. And then what the simplex method does, it stays entirely, it moves along the edges. So from here it will look to see in which direction would the cost go down, because we're trying to minimize the cost. So it would check these directions. Each of those directions we're releasing one equality. We're allowing one equality to be an inequality. and that moves us along.

So the simplex method has two steps. It checks each of these directions to find out which way will the cost drop fastest. It chooses the direction in which the cost, the gradient, the component of the gradient, you could say, along that edge is the biggest, or maybe the most negative. And then, once it decides which way to go, it goes -- maybe it takes this direction -- it goes, goes, goes, goes, goes until it hits another corner. So that's the end of the simplex set, when it reaches another corner. That completes one simplex step. Then, from this corner, it will look to three ways it could go here. Well, it's not going to pick this way because in this direction the cost was decreasing or we wouldn't have taken it. We wouldn't have taken that direction except for the cost went down. So if we came back, the cost would go up. No good. So going down would be one of these two ways. Maybe it goes down in this direction, so we decide on that direction. We follow it until we hit a new corner, and eventually we're going to get to the winning corner because there are only a finite number of corners. And how will we know it's the winning corner? Well, we'll

know that corner is a winner if in every direction the cost goes up. If the cost goes up in all directions along all the edges out of that corner, then that corner has won. It's the minimum. I'm using linearity here -- the fact that you know everything by traveling along, by looking along those edges.

So the simplex method is a big success. But in reality, in practice, it turns out that the number of edges that you have to travel to get to the winner doesn't grow exponentially. I mean in principle it could. People have dreamt up really desperate examples in which following the simplex method you could take a long time, but on average you don't, and in practice you don't. So it's a very good method and was totally the method of choice.

But a competitor has arrived. And that competitor goes under the name of interior point method, and you can guess what that method is doing quite different, totally different system. That method is inside the feasible set. It finds a point somewhere near the middle maybe. And then it does a normal gradient type approach from your point. It figures out which way to move. It moves, but it doesn't go outside. It doesn't even reach the boundary of the feasible set because if you reach the boundary of the feasible set you're out of the interior and the method is not going to operate. Well, that crudest method would be follow the gradient, but we know from several situations that gradient descent can be less than optimal. So this is more subtle. Well, Newton's method actually -- I'll explain.

So this is actually the content of my lecture -- this interior point method. And let me just mention a few names. People thought of interior point methods long ago, but a big splash came when Karmarkar proposed an interior point method and proved that it converged faster than simplex methods in some problems. Well, he said all problems, actually. His advertising of the message was pretty generous. The sort of claim that was around was ten times as fast as the simplex method generally, and it was on the front page of the New York Times, and I remember going to a lecture in Boston with lights, TV lights on and everything. Well, maybe his exact message now isn't so much used, but you have to give him credit for stirring up the whole world of optimization because the result of Karmarkar's method -- and there were others --

and I'll say barrier method, and that's what I'll try to explain. He stirred up the whole world so that the experts in optimization began looking again at interior point methods, seeing that they did have some merit, improving them. And now for, I would say particularly for large, sparse problems, these are a way to go. These are preferred now.

So this is the normal situation in scientific computing: that any message that's good, it's still not good for everything. It's got its range of problems where it's successful and a range of problems where some competitor wins. So that's the situation now. These are methods. This is certainly not out of date, and I'm sure it's the method of choice and it's carefully coded and well understood, but these are quite effective.

OK. So my job then is to say something about these interior point methods. And the beauty of these is that the primal -- this is called the primal problem. Primal problem. And often you write a (P) for primal. It means the given problem. And over here is the dual problem. So you put a (D) for dual problem. [UNINTELLIGIBLE PHRASE] It involves the same data: the same b , the same a , and the same c , but a new variable y . And [UNINTELLIGIBLE PHRASE] in the original problem for the constraints.

So I won't go at the dual problem exactly that way. I'm going to ask you just to consider this problem and show you the relation between the two. So what I want to say is that these two problems -- the primal and the dual, which use the same data a b c , are intimately related, and sort of solving one solves the other one. Actually is this it? That applies to the simplex method. When the simplex method finds the best corner, we could read off of Lagrange multipliers, we could read off y . We could read off the optimal y . So my picture was in the primal case, but there's a dual picture in the dual case.

OK. So we have a minimum problem and a maximum problem, and I'm using this word duality. So what I want to do is tell you how do we recognize the winning corner in the primal problem, and it's beautiful. So at the best, so the optimal -- x , let me call it x^* and y^* -- have \min over there equal of \max here. \min of all the

cx^* , which is cx^* , equal to a maximum over all of the y 's of the yb 's, which is y^*b . So these are equal at the winner.

That's the essence of this duality. Duality is about two problems that use the same data but they look quite different. You know they're using the data in different ways. The cost function there showed up in the constraint here. The constraint b there showed up in the cost function here. And even a got flipped, because if I use my usual column vector notation -- if I just transpose this -- this would be a transpose, y^T transpose, less or equal to c^T transpose. If I wanted to stay with column vectors y^T transpose and c^T transpose, then it would be the transpose of a that would appear. So I'll just put transpose with two exclamation marks.

That's typical. And you often see the word adjoint. So there are methods in differential equations, in optimization, called adjoint method. Adjoint is just really another word for transpose. It's a word that applies in differential equations as well as matrices, so it's kind of a better word, you could say, where transpose we usually apply to matrices, but totally the same idea, identical idea.

OK. So the wonderful thing is that at the moment of success, at the moment of optimality, these are equal. A minimum equals a maximum. And that's one way to recognize that you've succeeded, and that's one way to measure how far you have to go with the duality gap. So the duality gap would be the difference if you had a particular y that wasn't the winner, a particular x that wasn't the winner, the duality gap would be the difference between cx and yb . And what I'm saying is that when that duality gap narrows to 0, you've got it. When this narrows to 0, you've brought cx down as far as you could, you've raised yb up as far as you could. And if you did it right, if you've got to the optimum, then the duality gap disappeared -- became 0.

So that's a measure of am I at the answer, am I close, do you know if we're going to do an iterative method as I'm planning. So that's the point, of course. These interior point methods will be iterative. We said we never actually allow them to get to the absolute corner until maybe at the last minute.

So here, let me draw a picture of how interior point methods might work. So here is

the feasible set -- some kind of a polyhedron, whatever. So think of that as a kind of a diamond, a twenty-four carat diamond. OK? And start at a point inside. And somehow find a gradient, decide which way to move. And there'll be some barrier here which is going to prevent us from reaching it, so we'll stop. So that will be one step, and from here we will do the same thing, whatever it is. It'll be Newton's method, actually. You'll see. It's just Newton's method. The most fundamental way to solve non-linear equations is Newton's method. And it'll take another direction. Again it'll stop, and the thing will follow some path. And then maybe at this point the duality gap is very small. We'll realize that this is the winner. So we could at the last minute say OK jump to the winner. But it's this path through the interior that we're really interested in.

OK, so I'm giving a sort of general picture of it. And now I'm ready to do two things. One is the nice little bit of algebra that says that this duality gap is always greater or equal 0. OK so that's called weak duality. Weak duality, which is easy to prove, says that always cx , for any feasible x , is greater or equal to yb for any feasible y . So any x and y that satisfy the constraint. I should say satisfying the constraint. So weak duality I'll now prove in one second, And the point is that as I push to bring cx down -- minimize -- as I push to move yb up -- maximize -- they will meet at the winner.

OK, now how do I prove cx greater or equal yb ? Let me try to prove that. Proof. OK so look at yb . OK. Now so I know something about the constraints. Ax is greater or equal to b . So this b -- I want to say that this is less or equal to yAx . Now am I allowed to say that? First of all, y is feasible; x is feasible. So they satisfy. Feasible means that these are satisfied and these are satisfied.

OK. And do you see that that's really all right? Well you might say no problem. Ax is greater or b . It's obvious. But I have actually used one more point here, haven't I? If I have an inequality, then I'm multiplying it by y , and it didn't change the direction of the inequality sign, and that was because y is greater or equal 0. That's where that paid off. So this used the fact that -- this came from the fact that y was greater or equal 0 and Ax was greater or equal b . Those two facts meant that I could multiply and preserve the inequality sign.

And now I'm going to go to the next step: y^A is less or equal c , and there is x .

OK, so you see that I finally got what I want: y^b less or equal to cx . But what went into that step? Well looking here, I had y^a less or equal to c , and I also had x greater or equal 0 by the feasibility of x . So that inequality I was allowed to multiply by x because x is not negative. If x had been minus 1, then when you -- right, if I have an inequality like 4 less or equal 7, if I multiply by minus 1, I guess minus 4 and minus 7, and the inequality switches: minus 7 is below minus 4. But that's not what's happening here because the x is not negative, so this is not what's happening, and I'm OK, so the conclusion was exactly what I wanted -- that the y^b was less or equal to cx . And you see how perfectly it used the four inequality constraints.

OK. So that's the weak duality where the proof is easy. Just use what's given. The duality, without the word weak, is the fact that at the optimum the gap is 0, and actually we can see. That will tell us a lot. That will tell us a lot. When could this gap be 0? So at the optimum, y^* , equality is holding throughout. So if equality is holding, how can that be? How can I take these -- of course the inequality, the x^* and y^* are feasible. So if I just put stars on all these things, then I would have -- everything would still be totally true. But when I put stars on them, so I'm picking the optimal guys, then equality is holding. I still have these inequalities, so what I want to find is the optimality conditions. How are they related? If I have y greater or equal 0 and Ax greater or equal b and I multiply, how could I get equality? Right? For example, if I have 3 greater than 0 and 5 greater than 2. If I multiply those, I get 15 greater than 0, I guess, and that's far from equality, right?

So how could equality happen? Well, the only way is if one or the other of these, if equality holds in one or the other, then I would be OK. Yes, do you see that? If equality held -- these are vector inequalities, so I'm going really component by component. Let me write down the my conclusion and then you'll see what I mean. So these are called the Kuhn-Tucker conditions. You've seen their names before. And they're also called -- well long words -- complementary slackness. I'm using words that, if you haven't seen the subject, you think OK, who needs the long

words. But the idea of slack variable; slack is the difference. The slack is $c - y^T A$, or over here the slack is $Ax - b$. These are the slack variables. w , let's say, is $Ax - b$. And of course it's greater or equal 0 that's the nice thing about slack variables. You know you fix it so it's greater or equals 0. Here the slack variable s for slack would be what? $c - y^T A$ greater or equals 0. And there's no slack when s is 0. OK, so that's where the word slackness comes in. Slack is just the amount of give in the inequality.

So what's the point here? I was looking at this guy, and the only way that I could have equality here when I have inequalities there is for each component I'm going to have to have equality. And how can I have equality on a component? Well, I would have it for example if y was 0. Then when I multiply, I have equality. Right? OK. Or I could have equality if I had -- let's see, so what I want to say? So I want to say either y_i is 0 or $Ax_i = b_i$. Equality holds in one or other of the two inequalities, because then if I multiply them together, I have equality. Right? You see that. If one of those holds, say this one holds, if y_i is 0, then I certainly can multiply the inequality by y and I get 0 equals 0. Or if Ax is exactly b , then multiplying by y won't change. OK. So this is the complementary slackness, one or the other, that has to hold to get equality.

Now what about this guy? Equality same idea here. I got the inequality by multiplying these together. When will I get equality? Only if either x_j is 0 or the j component of $y^T A$ equals the j component of c . Again, the same reasoning: that when I multiply two things, if I get an equality out of two inequalities, then one of those two at least must have been actually an equals; otherwise I'd still have a gap.

OK, so this is pretty important. These are the conditions -- these are our equations. That tells us when we've won. So this actually holds towards the winners. It doesn't hold for all the other guys, but at the winner because things are equal here. They had to be equal at every step and therefore the Kuhn-Tucker conditions had hold at the winner. So they hold at this winning corner when we find it. So the simplex method chases corners, finally gets to a corner, and it would know it had got there by the fact that it couldn't decrease any more. And if you look at the algebra, you

would see that that tells you that the Kuhn-Tucker conditions are satisfied.

OK, so the only proof I gave was the weak proof, that cx is greater or equal to yb because that's the nice one. I've proved that for equality we'd need these.

OK, now I guess I'm ready for the method. I'm ready for the interior point barrier method that tells me how to compute. OK, so I'm at an interior point. What do I do?

OK, so here's the method; here's the barrier. I'll call it a log barrier. I'll solve the problem of minimizing cx . I won't solve the exact problem. I'm going to minimize cx , minus I think some little number times a barrier, which is going to be a sum of the logarithm of the x 's. This α is going to be a little bit positive. I'll take it smaller and smaller because this part is really -- it's that that I really want to minimize.

Right? That's the original problem. This is the cost. I'm adding something to the cost but I'd better just be sure that I've chosen the sign of α correctly. By the way, this is discussed now in the latest version of my Linear Algebra and Its Applications textbook, a fourth edition. Editions one to three of that book and others have described the simplex method, and now it was just natural to include the interior point barrier method.

OK, so why do I call this a barrier? Because if $x_{sub\ i}$ gets to 0, the log blows up. The log blows down I should say -- blows down to minus infinity. I'm multiplying by minus α , so I get positive. The combination blows up. It couldn't be the minimum, so you can see the minimum is never going to make it to x equals 0 because at x equals 0, the thing I have here is plus infinity.

So now I'm just going to use gradient method. I'm going to solve this problem with the constraints, of course, with the constraints, and set derivatives to 0. Now I have -- you know it's not linear anymore -- the winner is not at a corner anymore. It's somewhere in the middle. Calculus operates. I can set derivatives to 0. OK, so I want to do that. And of course I'm still inside this feasible set, so let me see if I can put down the equations and the constraints. OK, so I still have the constraints. Now, forgive me, but I've made a change to Ax equals b . I could have started with that as a constraint. I've made that change to Ax equals b . How have I done such a thing?

I'm given the problem with $Ax \geq b$, but I'm also given the slack variable $w \geq 0$. So I guess it's just a little trick that's not worth -- you could just take my word for it, a little trick. My new variable is the x 's and the w 's. m plus n variables: the n x 's and the m w 's. And now, put that together -- so can I just maybe do this over in the corner here? Before I start on this, I changed to a new variable that that'll be x 's and w 's. And that will be greater or equal 0, right? Because the x was always greater or equal 0, and the slack says $Ax \geq b$ is turned into slack greater or equal 0. And now that multiplies A minus i to give b , because Ax minus the slack variable is b , which says that Ax minus b is the slack variable w -- bring that over and that over -- and that's what we said was greater or equal 0.

Do you see that I've changed to an equation by introducing more variables? Putting the x 's and the slacks all together in a big variable that I'm now going to call x . So this is now the sum of the m plus n of these now because this is the new x and this is the new A .

You might say why didn't I just start with equality constraint, and I certainly could have done. But just to see that inequalities have their place too, and to see that we can get between one and the other. OK, so now this is the problem with equality constraint. So my new constraints are $Ax = b$ and $x \geq 0$. That's the primal constraint. And what's the dual constraint? So the dual constraint is $y \geq 0$. Right? And, OK I have to get this right because we're right at the end. And the slack, let me just write the slack one. The slack one is s is the slack. This is s . I'm going to transpose so that I have consistently columns vectored. So that when I transpose it says that $A^T y + s = c$. Right? I put that over there with the s and transposed to get column vectors. I like to have column vectors.

OK, so those are the constraints, but now what's the derivative equals 0 equation? Derivative equals 0 is the derivative of this equals 0. So what does that say? That says that if I set the derivative as a 0 that says that c_i -- the x remember is, well x has got all these components. c_i is α and the derivative of $\log x_i$ of course is $1/x_i$. So that's the equation for derivative equals 0. So this is

what I'm solving. OK Equality is here, equality is here, equality is here but non linear. This is of course non linear.

So Newton's method just says linearize. Newton's method is just linearized at the point, and that gives you the direction to move. And you move that direction because you've linearized. As you move, you're wandering a little away from precision, from perfection, but if you don't take too big a step, Newton is safe. Maybe since this is a course in scientific computing, I should've written on the very first day in big letters Newton, because that idea of following the gradient is the central method of solving non-linear equations. And then on the board beneath I would have written in big letters Carefully because the derivative is a local thing. And if you follow the derivative a long distance, follow the derivative here a long distance out to here, who knows what -- you've lost the safety of Newton's method. So Newton's method always comes in reality with some kind of a trust region, some region where you can rely on the derivative being a reasonable approximation of the way the function is moving. OK, so we do that here too. OK. maybe I won't write out in full notation. What does Newton's method do actually? So Newton's method we're at a particular x , y , s , and we've got to move. So the unknowns are --- the components of x , the components of y , and the components of s . So Newton's method takes steps: a Δx , a Δy , and a Δs , computes what those should be, and then that gives the direction, and if you take them exactly, that's the full Newton set, which you would be very happy to do because that gives terrific conversion, but if it's too big a step, then you have to cut back. So the equations for these are what you need so there'll be an $A \Delta x$ will be 0. Because b isn't changing. There will be a A transposed Δy , a transposed Δy plus Δs will be 0 because the c isn't changing. And then we'll get an equation out of this, which is a really significant one, but maybe time is running out on and I'm not going to do justice to. But that's a non-linear term, where you see if I keep $A \Delta x = 0$, then my new x is exactly feasible right? If I'm at an $Ax = b$ and I move it by a Δx that's in the null space, then I have -- All I'm saying is that when I take that step I will have $Ax + \Delta x$ still equal to b . Good. Constraints still satisfied. When I take this step, since that's linear, the constraint when I add on the Δy and the Δs

and the 0 I still have -- my new point still satisfies that constraint.

But this is of course not exactly satisfied. If I had the solution to this, I'd be done. That's my problem. Anyway, so it's not exactly satisfied. Newton would tell you a linearization of it, and you would move in that gradient direction to try to make the things -- to try to make equality hold, because our current x doesn't have equality hold. And of course the c is a transposed y plus s . So that equation -- You see what's going on here? This is a transposed y plus s , and the x is multiplying those, so there's a product there. And when I take the derivative it's a product rule. I get two terms. Anyway, I get a third equation from here for Δx that connects Δx , Δy , and Δs . I take that step and that's my interior point method. That's my Newton step.

So maybe I just end by reporting results, so I'll end with just two comments. First is, is the method any good? And of course you only know by trying. And the answer is yeah. Typically you get the gap -- you get the duality gap down below 10^{-8} , which is usually very satisfactory in 20 to 80 steps. You can never prove a statement like that because you can always create some awful example, but this is the typical performance of the method. Which is pretty good, regardless of m and n . That's what's wonderful -- that the number of steps doesn't increase with the size of the problem. Of course, the cost per step does increase with the size of the problem.

OK so that's the results, and that's why the method is popular. And now I just wanted to not leave duality, which is such a key idea, without going back to our much more familiar problem of quadratics, where there are quadratic terms. And the best model you remember was projection. You remember that we had a vector b and we have the line, the null space of A . This was the column space of A . This was all Ax 's. And perpendicular to it was the null space of A transpose. All A transpose y should equal 0. Do you remember this? This was the model problem for understanding. So that the projection of this solved one problem. The projection in the other direction -- we called that p . This was the projection p equal A times the best x . The projection in the opposite direction found the e , the error, but it was the

solution to the dual problem.

And now I want to say where was duality in this picture? Well duality was -- let me call it \hat{e} , the winning, the projection, the right guy over here. Or maybe \hat{y} . OK where was duality? Duality came in this case in the fact that it was Pythagoras. Duality in this simple, beautiful problem was simply the fact that p squared, this winner squared, plus e squared, was b squared. The winners were the orthogonal projection.

And now where is weak duality? It's the last second of the lecture. Weak duality says take something that's allowed, like that and take something that's allowed here, like that. And then those are not the winner. Those don't deserve stars or hats. They're not the winner. And compute that squared plus that squared. So this is any Ax . So any Ax squared and any y -- let's call that y squared. And what is the inequality that is satisfied by any Ax , like the wrong one here, and any y , like the wrong one there, will satisfy. Pythagoras won't be quite right. It'll be Ax squared plus y squared. What do we know about the sum of those two squares? It's greater than or equal to b squared. The only way we get this thing split into two orthogonal parts whose squares add up to b squared is right triangle. If I replace this by something longer and I replace this -- I should take that error really. e is really b minus cAx . That's what I should be putting here. This thing should be b minus Ax squared.

Anyway, the duality is in the fact of getting an equal sign there and weak duality is the easy inequality that no matter what you do, you get greater than or equal. So the duality gap is somehow the gap there, and the whole subject of optimization is to bring that gap to 0. So this is the gap in quadratic problems, of which this is a neat model and this was all about linear programming. And duality is present for both.

OK, so Friday is the promised lecture on ill-posed problems. And meanwhile if two people are willing to put up a hand now or email me later and say sure I'll take my turn Friday of next week, that would be terrific. OK. Thanks. I see one hand. OK.