ConvexOptimizationI-Lecture09

Instructor (Stephen Boyd):Okay. We'll continue on duality today. One thing though before we get started, I've received from a variety of sources, feedback that the homework is maybe just a little bit too much? I don't believe – I think it's fine, frankly, but – so the question would be – how many people would support, like, suppose we were to hold back. By the way, there's a limit to how much we can hold back. It's gonna be very mild. But suppose, I mean, the teaching staff, we could exercise a small amount of restraint in assigning the homework. Would there be general enthusiasm for this, or? Yes. Okay.

So let me limit – there's limits to our effectiveness. So we'll – but we'll see what we can do. What's the lower bound? Oh, no. No it's not very far. We have a very small gap between what's feasible and what the lower bound is. We might back off, like, one or two – one problem or two. Yeah. You know. Sorry. So that's what we'll – that's what we'll do. I mean the fact is this is how you actually learn the material, so don't see any reason to back off more than that, so okay.

Well, since I didn't hear, like, people screaming and yelling, that's sort of consistent with our information gathering, that it's – that some people think it's a bit too much. Actually, I got some complaints from faculty members in other departments who said that all research in their department had come more or less to a standstill because of this class, so. Interestingly, they did not advocate backing way off the homework, interestingly. They said back off a little bit. That's all. That was their suggestion.

Not that I'm under any obligation to do something a colleague in another department says. Okay. All right. Let's – you can go down. We'll talk about complimentary slackness. By the way, these – the topics we're covering right now involving duality – full optimality conditions, things like that, these are very tricky. So you actually have to go over these a couple of times to get the full exact logic of what implies what.

It is quite tricky. So just – just to let you know, actually I read this stuff very carefully a couple of times actually just so I can remember sort of exactly what implies what because it's kind of subtle. Okay. And let's see if I can end up not confusing myself today. Okay, so assume strong duality holds. Now what that means is we have a primal optimal point, X star, and we have lambda star, nu star, which are dual optimal. [Inaudible] these are just any primal optimal, any dual optimal point.

And of course what it means for strong duality, this means that the value achieved by X star, which is feasible, is equal to G of lambda star, nu star. Now that's a lower bound on the optimal value. So if you've got a point that's feasible and has an objective value equal to the lower bound, you're done. It's optimal. So you can think of lambda star, nu star as a certificate proving X star is optimal. By the way, the same is true the other way around.

X star is a primal certificate proving lambda star and nu star are optimal. Okay, so let's see what this means. G star of lambda star, nu star is by definition – it's the infimum over of this thing. I know we went over this last time, but I just want to sort of go over this again quickly. This is the infimum over X – over all X here of this lagrangian. Now, of course, that has to be less than or equal to the value of the lagrangian if you plug in X star.

You plug that in and you get this. But X star is feasible. But therefore, F I of X star is less than or equal to zero. The lambda I stars are bigger than or equal to zero. They have to be. That's dual feasibility. So this whole term is less than or equal to zero. This term is equal to zero because these H I's are zero. Therefore, this thing here is less than or equal to that because it's this thing here plus zero, plus something less than or equal to zero.

So it's less than that. Therefore, everything in between this chain has to be equal – every expression because if something is blah, blah, blah, less than this, less than that, back to something again, everything in between has to be equal, and that tell us, since that's zero, it says this thing's gotta be zero. Now that's a sum of a product of non-negative numbers with non-positive numbers. And that zero – that happens only if every single product is zero, okay?

Let me just – today this'll make sense. Last time it was just an algebraic fact. Today there'll be lots of interpretations under which this makes sense. What that says is the following – for each eye – for each constraint, either – lambda I star is zero, so either the optimal lag range multiplier is zero, or the constraint is tight. That's what F I of X star equals zero means. So you either type – or the constraint is zero.

And you can – logically you can twist the logic around and say it many other ways. You can say things like this, you know, if lambda I star is positive, then the constraint is tight. If a constraint is slack, that means F I of X is less than zero, then you must have lambda I star equals zero. And later today we're gonna have some interpretations of lambda I star that will make this completely clear and totally intuitive.

So this is called complimentary slackness, and what it really means is that the vector of these Fs and the vector of the lambdas actually have complimentary sparsity patterns. So a sparsity pattern in a vector tells you what entries are zero and what are non zero. And this has says they're complimentary. They cannot have non zeroes in the same entry, so that's the picture.

Okay. This brings us to something called the KKT conditions. They're some fun history here. Kuhn and Tucker are sort of well known – who had worked these out at some point. I mean these are kind of silly because I think, for sure, there were people in Moscow for example who know these earlier, but it doesn't matter. And then actually somebody uncovered a 1938 master's thesis. This poor guy Karush – that had everything in it. And actually, miraculously, they added his name to this.

So these are now – in the west it's called the KKT conditions. So this is basically a generalization of the very simple fact that for an unconstrained problem, the necessary and sufficient conditions for optimality, or a convex function, is if the gradient is zero.

That's it, period. You already know that. Now the question is what about with inequality restraints and so on. And these are for differentiable – differentiable F I and H I.

Okay. Here it is. So here it is. This is – you have to have the following. Obviously, you have to have primal feasibility. So X could not possibly – by definition it could not be optimal if it weren't feasible. So feasible means that it satisfies the inequality constraints and the equality constraints. That's feasible. Number one. Dual constraint – dual feasibility says that the lambda Is are bigger than or equal to zero. Complementary slackness says lambda I F I is zero, and then the final one is gradient of the lagrangian with respect to X vanishes. That's this expression here.

So this is simply – you might write this something like this, you know, partial L, partial X, something like that. That's a gradient. I guess it would be all right this way. Gradient X L equals zero, okay? So these four conditions together are called the KKT conditions. And, in fact, what we just showed is the following: if you have a problem – by the way, convex or not, either way, and strong duality holds, then the KKT conditions will hold for any pair of a primal and a dual feasible – sorry – a primal optimal and a dual optimal – here, got a question? Oh, okay. Sorry. Okay.

So that's the – so if strong duality holds, the KKT conditions must hold, okay? Now, for convex problems, it's necessary – it's sort of necessary and sufficient, and you have to get sort of the logic exactly right. So let's see what that means. Now suppose you have a convex problem and some points, X tilde, lambda tilde, and nu tilde satisfy KKT, okay? I'm writing them as tildes because if I put a star on them it sort of presupposes by the notation. It implies that they are optimal.

They will turn out to be optimal, but for right now I'm gonna use tildes, which are more neutral, okay? So suppose – now X tilde, lambda tilde, nu tilde satisfy the KKT conditions. Then we're gonna argue that, in fact, they're optimal, okay? X is primal optimal, lambda nu is dual optimal. Okay, so the way that works is this – from complimentary slackness – if complimentary slackness holds – if all of these conditions hold then, in fact, lambda I F I is zero, right, because for each I, either lambda I is zero or F I is zero, okay?

These are – all – the other things, the nu I H I – those are all zero, and therefore, L of X tilde, lambda tilde, nu tilde, is equal to F zero of X tilde, okay? So we'll start with that. Now the fourth condition, that's this gradient condition here. If this holds, then look at this. F zero plus lambda – sum lambda I F I, plus some nu I H I is gonna be a convex function. Why? Because the H Is are affine. You can multiply them by any number and still have an affine function. The F Is are convex. I can multiply them by any non-negative numbers, still get a convex function. I can add up a bunch of convex functions to get a convex function differentiable.

This condition says that the convex function F zero of X plus sum lambda I F I plus sum nu I H I, that this thing has a gradient that vanishes. If that has a gradient that vanishes, it's a convex differentiable function, then at that point, that point is the minimal.

Therefore, that says that X tilde here minimizes – is a minimizer of the lagrangian, and therefore, if I evaluate L of X tilde lambda tilde nu tilde, I get G. Hey, I'm done because now I've just shown that these two are equal. That's an objective – that's obviously upper bound. On the optimal value, that's a lower bound. They're equal. It's optimal, okay? Now there's also the question how Slater's condition comes in, so let's review what we've seen. We've seen the following. If for any problem, convex or not, that you have a strong duality of [inaudible] – If that's the case, then the KKT conditions hold. And the converse is false. Actually, in other courses, not this one, you'd study that like crazy, and you'd have numerical methods that would search for KKT points, and there'd be false KKT, you know, KKT points that aren't optimal, and so on. But as far as we're concerned, any problem you have strong duality, KKT conditions hold. That's number one. Number two, if the KKT conditions hold for a convex problem, you have – then those points are a primal dual optimal pair, period. Okay? And the last one says this. You have to have Slater's condition. Slater's condition says that if Slater's condition is satisfied – that means there's a strictly feasible point – there exists a strictly feasible point, then it says the following. It says that a point is optimal of, and only if, there exists lambda nu that satisfy KKT conditions. So that's the full logic of it. Question?

Student:

When you say [inaudible], do you mean [inaudible]?

Instructor (Stephen Boyd):No, no, no. What I'm saying is – just let's go very – so here, the KKT conditions – I mean you have to be very careful about how you state it to say it's necessary and sufficient, so I'll state one carefully. Ready? It says this. If a problem is convex – in fact, I'll just say the true statements. Ready? So the trust statements are as follows. If [inaudible] assumption on convexity. If X – if a problem has strong duality obtains for it, then the primal and dual optimal points will satisfy KKT conditions, period.

By the way, this is very much like saying when you minimize a non-convex differential function, at the minimum, the gradient will vanish. That's true. The converse is false, of course. Same here. Okay, so that's the analog of that classical calculus statement. Then the next one says this. If the problem is convex, and a set of points satisfy the KKT conditions, then in fact, they are primal and dual optimal. So at that point, if you wanted to, you could say something like this.

For a convex problem, the primal dual optimality is absolute if, and only if, KKT holds. So that's one way to say it. Now if you wanted to have a one-sided statement to say what's the condition – I mean, you know, if you think about it, all of the duality stuff we've constructed is sort of – well, we've constructed it. I mean after a while it will become quite natural, but it's sort of like the [inaudible] transform. It's not – it's a cognitive construct. We made it up.

So after you start thinking in the [inaudible] transform, then someone is allowed to give you an answer that involves the [inaudible] transform. Once you start thinking about duality, which you will, I promise, in the next couple of weeks or whatever, then it'll be natural for me to just say the statement I said. However, if you just say, look, I don't care about – what's this duality? I don't even want to hear about it. I have my optimization problem and I want to know when is X optimal.

Okay? So if you wanted a one-sided statement that only focuses on X, then it has to be this one. It would be something like this. X is optimal if, and only if, there exists lambda nu that satisfy KKT conditions. That is almost necessary and sufficient conditions for a convex problem. It's not quite because you have the possibility that, for example, Slater's condition doesn't hold, and you have positive duality gap and all that.

Student: Where does convexity enter in?

Instructor (Stephen Boyd):Do what? What I just said?

Student: Yes.

Instructor (Stephen Boyd):When you remove the convexity assumption, all statements go away except the necessary ones. Right? So it's exactly like sort of going back to calculus or whatever and gradient vanishing. So it's got the same status, works the same way, everything like that, so.

Student: But I mean how did you get that [inaudible] condition?

Instructor (Stephen Boyd):Where'd I get what?

Student: Where did you get this statement?

Instructor (Stephen Boyd):Which one?

Student: It's in parentheses, the one [inaudible].

Instructor (Stephen Boyd):Here?

Student: Yes.

Instructor (Stephen Boyd):Oh yeah, I'm just saying, look, the fourth condition is this, provided it's a convex problem means F zero's convex, F I's convex, H I's affine. Also, lambda I are positive. Therefore, this function here is convex. Gradient of a convex function vanishes, differentiable, done. It's minimized. So that's the logic there. No – I mean there's a lot of – I mean it's not totally straightforward – all the logic here. I believe I have now confused everyone. I don't see anyone who's not confused. So that's a good time to move on.

Okay. So there are some examples of this. Let's do a quick example of an application of this. Is water filling – this comes from communications and information theory, but actually there's problems like this all over the place. In fact – well it doesn't matter. But we'll just look at it as a canonical example. And let me just explain what it is. You're maximizing the sum of the log of X I plus alpha I. And I can even sort of – I can give – I mean I won't do a good enough job to explain to people who don't know about this, but for those who do, I'll connect it.

Log of X I plus alpha I, here you're doing something like you're allocating power to a bunch of transmitters. By the way, if you don't care about communications, ignore what I'm about to say. You're allocating a total amount of power, say one watt to a handful of transmitters, or across different channels or something like that in the communication system. Log of a constant plus a signal to noise ratio is actually gonna be your – is proportional to your actual bit rate obtained on that channel.

So this says maximize the total bit rate by allocating some power across some channels. Okay? By the way, this is used, for example, now in DSL. It's used, actually, everywhere, okay? And I'm not talking about used by professors, I'm talking about, like, it's used when you use DSL for example. So – okay, so otherwise, for everyone else, it's just, you know, it's just maximize the sum of the logs of X I plus alpha I. Subject to this.

And you can even get some intuition about how this is gonna work. Here, you know, obviously log is concave, so you have sort of diminishing returns that if you look at how much power you put in, or, you know, how much X I you allocate here, if alpha I is small, then putting some X I into that channel is gonna give you some – a very nice rate. Once alpha I is big, the incremental return is gonna be very small, right? So it's kind of clear that you want to put the – you want to allocate X because we're allocating a total of one here.

You want to allocate a bunch of the Xs to the small alpha Is. I mean that's clear. Exactly how much – how to do this is not clear. But we'll see how that works. Well, if you write the KKT conditions this is very, very simple. It's this. It's that X of course has to be primal feasible. That means it's a probability distribution. It adds up to one, and it's non-negative. There is a – we have a lag range multiplier lambda associated with this – these inequalities. And we have a lag range multiplier nu associated with this equality constraint. And you just write out what it is.

I mean you write this thing plus sum – actually minus lambda transpose X. That's the term because this term, when you want to put it into lagrangian, becomes minus to flip it around to make it negative. It's minus X is less than or equal to zero. So this is minus lambda transpose X plus nu times – nu is a scaler – times one, transpose X minus one, okay? Now you differentiate that with respect X is separable in X. Right? The terms – it's completely separable in X. So partial L, partial X I is zero, is this. It's nothing but that.

Okay? Then you have the complimentary slackness. It says lambda is – lambda I is zero if X I is positive, okay? Assume – by the way, we'll get later this lecture to what lambda I is, and you'll have a beautiful interpretation as in communications as well. It'll work out beautifully here as to what lambda I is. Okay. Now let's – so this is the KKT conditions.

They have to work here. By the way, the stronger form of KKT conditions says this. Linear inequalities sort of get a free pass in the strong form of KKT conditions.

So weak form of KKT conditions says there must be a point that satisfies the inequalities strictly. Now of course there is here. Obviously there is. You just take X equals one over – X I equals one over N. So the strong form of KKT holds here. But you don't even need that because if the inequalities are linear, they sort of get a free pass in Slater. Okay? They go back to just mere feasibility. So – okay. So if that Slater holds so everything's gonna work. Strong duality obtains everything. Okay, so the conditions are this. Lambda's bigger than and equal to zero, complimentary slackness, and this is this gradient condition. But let's take a look at this.

That's non-negative, okay? So let's actually try to figure out what happened – if this is non-negative, it says nu for sure is bigger than one over X I plus alpha I, period. Okay? Now one over X I – sorry – nu is less than or equal to because you add a non-negative thing to get nu. So nu is less than or equal to one over X I plus alpha I, but whenever X equals alpha I of X I goes between zero and one, it varies between, you know, one over alpha, and one over alpha plus one.

So, in particular, if nu I were less – if nu – sorry, if nu is less than one over alpha I, then here you absolutely have to – you must have lambda I equals zero, period. And in effect you get X I. X I is one over nu minus alpha I – just by solving these equations. I'm just arguing from the KKT conditions, okay? Now if nu is bigger than one over alpha, then lambda I has to be equal to this, period, and you have to have X I is zero. Actually, the argument really goes like this.

If [inaudible] you have to have X I equals zero, and therefore lambda I is this. So, okay, but now you have an explicit formula for X as a function of nu. Nu is a single scaler variable, and you basically have this. You have – dual – primal feasibility says ones transpose X is the sum of this because you get a formula for this – for what X I is. It's just max of zero and one over nu minus alpha I. That sums up to one. This function here is monotone increasing in nu.

And therefore it has – there's a point where this is equal to one. So, by the way, this is sort of a classic example – this is what you'd find kind of in a, you know, to me this is kind of old style calculus or something like that. But I give it as an example. Let me tell you why I call it old style calculus. It's one of – what is calculus? Calculus goes like this. You say, look, here's what a derivative is, here's what a lag range multiplier is, here are these – you write down all these conditions. It's like differential equations or something, and it works like this.

Then in that class, they show you 12 of the 17 total cases known to mankind where you can actually solve those equations. And you know what I'm talking about here? Right? So you have a differential equation, and you look at 12 of the 17 that you can actually solve. No one who actually does anything solves things analytically. No one has done that since maybe the 1940s or something like that. I mean some fantasy from the 18th century, or something like that that persists, okay?

So a lot of people – they show this as, you know, they say, look, isn't this – so you can do it analytically. Actually, this is all quite silly because, for example, I just add one little minor variation on this, and in fact, you can't do it analytically anymore. But the computational complexity of solving it is insane. So as I just mentioned, this is kind of a classical one where you can actually write down the KKT addition and solve it. That's really not the point of the KKT conditions. I just wanted to emphasis that.

We'll see later in the class how, you know, people – it's actually even funnier than you think because people say things like no – defending so called analytical solution they'll say things like, oh no, no, but an analytical is much faster. Actually, that's false in fact. That is just completely false. Actually, computational complexity, if you use – if you solve this problem by methods later in the class, will be faster than, in fact, doing it by the so-called analytic solution.

Student: Do you need to solve the problem for every single alpha I? It seems like [inaudible] –

Instructor (Stephen Boyd):Alpha I are problem data. They're just problem data. So, in fact, they are the only problem data here. So to solve this problem means to be given a set of alpha Is, do some thinking, and to return the optimal allocation X I. So in that sense, yes you do.

Student: Why? Because it has something to do with comparing it to one over alpha?

Instructor (Stephen Boyd):Yeah.

Student:[Inaudible] is just a single number of alpha I since it wouldn't vary a lot.

Instructor (Stephen Boyd):It does.

Student: So which one do you compare it to?

Instructor (Stephen Boyd):Well, in each of these, you're comparing it to one of them, and then you add those up. So that's how that works. The interpretation, which gives us the name water-filling, is this. So what – there's a way to interpret this very nicely. What you do is you make heights alpha I like this, and then you pour in some water here. So you flood this area with, say, a unit of water. Or actually, in general, you flood it in the communication context with – this right hand side is P, which is a total power that you're gonna allocate to some channels.

You flood it, and the height – the depth of the water gives you the optimal X, okay? So that's the picture here. And it's actually kind of nice. It says that basically if alpha is high enough – if you become an island, then you allocate no power. So if you do communications, it makes perfect sense. It says if the signal to noise ratio is really bad, here’s the amount of power you allocate to that sub band or whatever – zero. It just doesn’t make any sense, okay? If alpha I’s really small, you’re gonna allocate power to it.

And the amount you allocate goes something like this. Okay. So that’s an example. One of the – by the way, handful – very small handfuls of examples where the KKT condition actually leads to a solution of a problem. Right? So, okay. Now we’re gonna look at – actually, a very useful interpretation of lag range multipliers, and in fact, it is what makes lag range multipliers extremely useful in practice. And this part is generally not emphasized. Certainly people know it – what is – it’s actually shocking how useful this is in a lot of practical context.

So here’s what it is. Let’s take our original problem – that’s this thing here – and what we’re gonna do is the following. We’re gonna perturb – notice the period. Everything’s cool here. See that – min means – it stands for minimize. That’s what the period means. Okay, so – [inaudible]. Everything’s cool here. So what you do is perturb the problem. And by the way, this is sort of a multi-objective. It’s sort of like a multi-objective multi-criterion problem or something like that. But you talk about – in fact, perturbation analysis of a problem is to look at the solution of this problem as a function of nu I – U I and V I.

Actually, that’s extremely useful, and actually, generally, should always be done. No exceptions. In fact, we’ll see it’s – generally speaking, it’s locally done for you when you solve the problem whether you like it or not. We’ll get to that in a minute. But – so let’s actually just stop and say what this perturb problem means. If U I is zero, and V I is zero, this thing is the same as that. Now, if I increase U one – if I make U one equal to point one, for example, then the way you’d say that is you’d say you have relaxed the first constraint because, well, you increased the feasible set. So you’ve relaxed it. Okay?

And it has lots of interpretations. F one less than or equal to zero might have been a limit on power in a circuit for example. When you relax it – F I is less than point one, it might have said that you just threw another 100 milliwatts at a circuit. Okay? So that’s sort of the idea here. It could be a constraint on anything, okay? And you just relaxed it. All right. Now if U I is minus point one, you’ve restricted – you’ve tightened the first constraint. So you should think of U Is as loosening and tightening knobs for the problem. Now, of course, if you tighten – if this thing is feasible, and you tighten this, it could be unfeasible. In which case, B star goes to plus infinity the optimal value, okay?

You can’t interpret the V Is as loosening and tightening. Actually, what they are is you’re shifting. So you’re really saying, you know, for example, in a quality constraint – might be an optimal control problem that you hit a certain target. If you mess with the V Is, you’re just changing the target point, okay? Is it loosening or tightening? Doesn’t make any sense. You’re actually shifting the – you’re actually shifting the affine constraints in the feasible set. So it doesn’t correspond to tightening or loosening either one.

Okay. Now what we do is you look at P star of U V. That is the optimal value of this problem as a function of U I and V I. By the w ay, that's a convex function in U and V. You know that. Okay. And it's really interesting – by the way, it's very interesting to say what this – this is a very interesting thing to look at, and I should mention a few things. If you were just take one U one, and say U one, and plot U one versus P star, you'd get some convex curve, and that of course would be the optimal tradeoff of F one versus F zero. That's exactly what it is.

Okay. Now – so P star, this gives you the optimal value, and you shouldn't confuse what it is. It really means the following. It means, for example, if I were to allocate you another 50 milliwatts of power for your circuit, and – this is the important part – and you were to reoptimize the entire circuit to use that extra power, then this is how much better you would do. Okay? That's very different from a different type – another sensitivity says what happens if you wiggle this but you don't reoptimize?

That's a different kind of sensitivity. This is sensitivity in design. So you – you're given something – or if something – if someone takes something away from you by tightening a constraint, you completely reoptimize and then you evaluate what's happened. And because, in fact, someone could take a lot of a resource away from you, and you could design around it, which is to say, you could reoptimize everything else, even though they've taken a huge amount of resource away from you, you're not doing much worse. Everybody see what I'm saying?

Okay. So – and we're gonna see all of that and how it all works. Okay. So this is the perturb problem. Now the dual of this is in fact just that. You can just work it out. So here it's maximize G of lambda nu. Here it's maximize G of lambda nu minus U [inaudible] actually it's extremely simple to show – minus U [inaudible] minus – because this – what was a zero here becomes a U. I mean it's very simple. Here, you get this? Okay. Now what we're interested in is what can you say about P star of U V? What can you obtain from the unperturbed problem and a dual optimal solution?

And so you get these global sensitivity results. You're gonna do something on this on your current homework. Some of you may already have. But maybe not. You all will have, I think. Actually, I can't remember. Is it this homework? Okay. We're obviously – we're deeply pipelined. We're working on your homework two weeks from now. So the homework you're working on is in our distant, distant and hazy memory. Okay, so you have the following. Assume you have strong duality for the unperturbed problem, right?

And lambda star nu star are dual optimal for the unperturbed problem, then you have the following. P star of U and V is bigger than G of lambda star, nu star, minus this. So you get these – just this affine thing. Okay? And let me just quickly draw a picture just to show you how that looks. Here might be U, as you change U, and here's P star of U, and it basically says that there's an affine function here that lies below C star of U. I mean it kind of makes sense. This is it. Now this condition is really, really interesting. So this star – sorry – P star of zero zero, that's the optimal value of the unperturbed problem. And so it says the following. It says if you tighten or loosen a constraint, or change an equality constraint in a problem, it says it's – actually, this doesn't require convexity here, but it doesn't matter we're gonna be interested in this in the case that it's convex.

Then it says that this optimal value as a function of U and V. You are the tightening parameters near the shifting parameters. It says it's – the change, if I were to subtract that over there, it actually – it's pessimistic. Note that this is an inequality constraint. So it says – what it guarantees you is you will do worse than the following. Worst means you'll be at least this value in optimal value. You will be at least as bad as minus sum U I lambda I star minus sum V I nu I star, period.

It says you will do at least that bad. You might do much worse, including possibly the thing will become infeasible and this becomes plus infinity. Okay? That's infinitely – that's infinitely bad. And now this means – this is a global result. This doesn't hold for U and V small. This holds for all U and V, okay? So okay. Now because it's this global thing, and it's an inequality, basically it's a pessimistic result, it actually is asymmetric. It makes weird asymmetric predictions about what will happen.

It says – basically there's cases where it says the following. Let's look at this. Let's look at a large lag range multiplier. You have a large lag range multiplier, then it says if you tighten that constraint, you are absolutely guaranteed that P star has to go up. Period. It may go up even more. It may go up to plus infinity. That's as high as it can go, meaning you've tightened that constraint and it's become infeasible. But it guarantees sort of in a minimum amount that the performance deteriorates, okay?

What's – interestingly, if you then take lambda I – if lambda I starts large, and you loosen it, you might think, oh, great, then my objective will go way down. And that's actually false. It makes no statement. It might not even go down at all. Okay? This is an inequality. And so you can work out various things here. That's what these are. You just have to look at them and see if you believe them. Now this is a global result. You can also look at this locally.

And you can – let's consider the case where this optimal value of the problem as a function of the perturbation parameters is differential. In that case, you have unbelievably simple formulas, and here they are. And this is it. I mean this is – that's the formula. That's what you want to think about. That's right there. So lambda I star is the partial derivative of the optimal value with respect to U I, period. With a minus sign. Okay?

If it's differentiable – by the way, it is often the case that P star of U V is not differentiable. In which case, the right hand sides here have no meaning. Okay? I mean the global result always holds. Always. This local one requires differentiability. But in terms of interpreting it, it's just beautiful. And let's actually just talk about what it means. Actually, let's look at complimentary slackness. Ready? Let's do this. Let's solve an optimization problem, and let's suppose that F three of X star is equal to minus point one. Okay? What does it mean?

It means that the third inequality constraint is slack. It’s not tight. Okay? Now let me ask you this. Suppose I go back to my original problem, which is this, and suppose I perturb the zero. Matter of fact, let me tighten it, or let me loosen it. Let’s loosen it to plus point zero one. How does P star change? Not at all because the same X star is optimal. So P star doesn’t change at all.

Can I – I can tighten it. By the way, I can tighten it by, for example, point zero one. I can replace zero with minus point zero one. If I replace zero with minus point two, what can you say? Actually, we can say very little, other than it will get worse or stay the same. Okay? So it could actually stay the same, and everything in between is possible. Okay. What is lambda three star?

Lambda three star is zero. It has to be zero. And you can – well, first of all, you could just say, look, complimentary slackness. Look, minus point one times lambda three star has to be zero. That’s complimentary slackness. Obviously, lambda three star is zero. Okay? But in fact, you get to the sensitivity. Lambda three star equals zero means you can wiggle the right hand side of that constraint and it will have no affect whatsoever on the optimal value. Everybody understand that?

And it’s completely clear. Okay? So this makes – so now you know what it means if you have a big lag range multiplier, optimal lag range multiplier, or small. Big means that constraint – first of all, it better be tight. Has to be tight, okay? It better – it’s tight, and it says that constraint is – well, let’s see. So far, we’ve had a qualitative idea of tightness of a constraint. You simply solve the problem, and you look at a constraint – an inequality constraint. If it’s equal to zero, we say it’s tight.

If it’s less than zero, we say it’s slack, or not tight, okay? You now have a numerical – a quantitative measure of how tight it is. Right? This is fantastically useful. Let me explain sort of in practice how this works. You know, if you start doing this, what’ll happen is you’ll have a whole bunch of constraints. And you’ll solve a problem, and we’ll look at – let’s just look at – you look at F one of X star. You look at F two of X star, and so on. And you’ll look at F 10 of X star, okay?

And these might be – let’s put down some things these might be. Well, let’s suppose that all of these are actually zero, okay? What does it mean? This means that when – at least, for the solution you found, X star, all of these constraints are tight. You know, that’s your constraint on power. That’s area. You know, I don’t know what this is. Just some timing – they’re all tight. Okay? Now by itself, this is like – I mean it doesn’t tell you that much. If I write down the optimal lag range multiplier, things are gonna get a lot clearer.

So suppose I write this, right, and this is 10. Oh, this doesn’t matter. Let’s make this three. There you go. Right? And this one is point zero two. Now what does this mean? We know that the first constraint, second, and third, are tight. Okay? And if you solve these numbers – and I don’t want a precise mathematical statement, I want just what would this mean in the context of solving that problem. It has a very strong meaning. What does it mean? And I want just a rough statement. What would you say about this?

Student:[Inaudible]

Instructor (Stephen Boyd):Right. It says that the set – you would say something like this. Someone would say, well are all the constraints tight? You'd say yep. But the second one – I – this sounds weird – is much tighter than the other two. I know that sounds weird. It's a violation of, you know, it'd be like saying is that matrix singular. Right? And then if you can – if you're – in certain context you're allowed to say things like almost, or nearly, or effectively. Okay? And that means you're talking about singular values, or a small – or a very large condition number. And it makes a lot of sense. You can say the same thing here. Okay?

And what you would see here is that, for example, F two is the constraint. If someone were gonna give you some more resources, and you wanted to ask for more resources, you'd actually ask for F two – whatever F two represents, that's the resource you'd ask for. If someone said we're tightening budgets, take something away, then you'd say, okay, please let's make it resource one, for example. By the way, doesn't guarantee anything. Okay? Nothing is guaranteed. What this says is that this right hand side's zero in the inequality can be wiggled a little bit, and at least locally, I can wiggle a little bit and have essentially no change in the optimal value. By the way, that does not mean X star. You have to do a complete redesign. So in fact, the correct interpretation is you can wiggle U one a little bit. And the redesign will work around it. The concept of the redesign is actually extremely important because that's what this kind of sensitivity is. So basically, someone takes away a little power from you and you're like, okay, no problem. And you redesign the whole thing to basically achieve all of the other specifications. It's very little worse, but it now takes up less power or something like that. So that's the idea. And so this is just, I mean this has huge implications in practice, and in a lot of cases it's just not – well, it's a disconnect, right? Anyone who knows anything about optimization knows about these, and I – almost universally, people who do optimization in application context don't know about these. So there's – it's like a perfect disconnect here. Okay, so now you have – this is sort of – this is the correct interpretation of lag range multipliers, okay? And in fact, if you go back, you get beautiful, beautiful interpretations, for example, here in water-filling. The lambda I now have a beautiful meaning. Lambda I is grade. It's basically – you're violating this constraint. That's the same, by the way, of sort of reducing alpha I. Sort of basically says, you know, if I were allowed here to sort of reduce my – that's something like a noise level, or something like that. If I were to reduce my noise level, how much more net total bandwidth would I get for the same power? And that's exactly what lambda I is. Gives you the partial derivative of that. All right? So it's literally in, you know, bits per second per watt is the unit of lambda I here. Okay. All right, so this finishes this – other interpretations of duality – they're in the book. You should actually read this. I mean just read them because they're just supposed to give you a feeling for what it is. It's very important you have a good. I mean, in addition to knowing all the math, have a good feeling for what these mean. They're – also they generally have interpretations of things like prices. And these examples are given. There's also examples in mechanics and other things. You need to look at these. We'll see lots and lots of examples of these. It'll come up in all sorts of context.

Student:

[Inaudible].

Instructor (Stephen Boyd):

Yep.

Student: From lambda two, or from F two, did you mean decrease – I mean just decrease the right hand side of [inaudible]?

Instructor (Stephen Boyd):That's exactly what I mean, yeah.

Student: Did you go negative?

Instructor (Stephen Boyd):Yep. You go negative. That's heightening that constraint. And here you're guaranteed – I can tell you that if you tighten U two for this problem, I can give you a lower bound on how bad things can get. You can tighten U two and P star can go to infinity, which is to say it's not feasible. So I can't give an upper bound if I tighten U two. I can give you a lower bound though. If I tighten U two to minus point one, for example, then someone tell me here how much P star is gonna go up by at least.

Student: Twenty.

Instructor (Stephen Boyd):Twenty. Precisely. It'll go up by one. Could go up by plus infinity. If you tighten U two to minus point one, and this thing becomes – the problem becomes infeasible, then it went up by plus infinity. But it has to go up by one, period. So that's the picture. Okay. Now let me also put this in perspective. It's gonna turn out the following. More – I mean, for the things we're gonna look at this quarter, not next quarter, but for this quarter, although duality plays central role next quarter, but when you look at them – sorry, I know you're not thinking about next quarter – as a [inaudible] order statement, the following is true. When you solve a convex problem, you're gonna get the optimal lag range multipliers whether you like it or not.

It's just – that's – you, in fact, will see all this in a few weeks. I mean that's basically how they're solved. I mean – so – and you've heard things like primal dual solvers, or you may have heard of this, or whatever, but it doesn't matter. Even if it's not a primal dual solver, whether you like it or not, you will not be given just X star. You'll be given X star, lambda star, nu star, and that's universal. So any solver is gonna do this, period.

You may not care about lambda star nu star. You may only care about X star. Fine. But the point is they're there. Maybe they're not being returned to you. Maybe they're being returned to you and you're ignoring them. They're there. What that means is basically these optimal sensitivities, if you want to call them optimal sensitivities, they're free, they're free. You design a circuit, period, you get these optimal sensitivities.

You can ignore them. Fine. No problem. But the point is they're absolutely free. They're just there. You know, you solve a maximum likelihood problem, these things are there. You do experiment design, these things are there. You do finance, they're there. So actually that's why, actually, all of everything – all the duality we're talking about actually has huge applications because this is all – this all comes for free.

These are not – there's no additional cost to calculating these. You get these whether you like it or not. So I – seems to me the right policy is when something comes to you for free, you should figure out some way to use it. And they're just fantastically useful just to look at. So this is a matter of course. And if you look at any solver – and I mean, like, a low level solver, one that solves an LP, a QP, and SOCP, and SDP, mathementropy, any of these, absolutely no exceptions, it will return dual feasible points.

I mean that's just part of the deal. Actually, if for no other reason than to certify the solution, right? Because that's what's considered in convex optimization what it means to solve a problem. You don't just say here's the answer I think. You say here's the answer, oh, and there's no reason you should believe me, so I'm also providing optimal lag range multipliers, which in fact form the certificate proving that what I have alleged as the answer is in fact the answer. Everybody see what I'm saying?

So that's just sort of the – that's sort of the social contract you have when you solver for convex optimization. That's what you get. Right? So – and then there's no questioning. I mean there's no questioning. I mean there's no – then – by the way, it's none of your business how they solved the problem because they've returned not only a solution, but a proof, a certificate proving it's the solution, and then it's none of your business.

They could have figured out X star by using a stick or something like that. It – but, I mean look. It's fine. It doesn't matter. If the certificate checks out, that's the end of that. That's the end of the discussion. Okay?

Student: How do you feasibly check certificates in your hand?

Instructor (Stephen Boyd):You evaluate G. So you evaluate G, and you – I mean, first of all, if you really want to – if you want to do this in a way where you don't trust the solver, it's extremely simple. Somebody gives you X – let's call it X tilde. We'll take a tilde off and put a star on it only after we certify it. So they return to us X tilde, lambda tilde, nu tilde. Okay? Tilde because it hasn't been verified yet. Right?

So we're doing safe optimization or something like that, right? So the first thing I do is I take X tilde, and I check if it's feasible. If it's not feasible, I leave a bad review for that solver. I go back and leave some customer feedback and say, come on, this is – they return to me a point that was supposed – they alleged to be optimal, it wasn't even feasible, okay? So if it – if X tilde is feasible, I think check that lambda tilde is bigger than or equal to zero, okay?

If they have a negative lambda tilde, I give an even sillier – I give an even worse review of that solver in a reputation system, okay? If lambda tilde is bigger than or equal to zero, now I go back and I check the KKT condition with the gradients and whatever – the fourth KKT condition. If that derivative – if that gradient is not zero, then I say come on, please. Stop messing with me. Okay? If it's zero, then I know what G of lambda star nu star – sorry – G of lambda tilde nu tilde is.

If that number is equal or very close to F zero of F tilde – if it's within my threshold, it's certified, end of story. That's it. And I don't – I leave positive feedback for that solver. Okay? So that's how you do it. Okay. All right. Next topic is really interesting, and let me explain what it is. There's a question.

Student: So for [inaudible] you just skip the differentiability step? You go straight to evaluating G?

Instructor (Stephen Boyd):There is a direct variation extension of KKT conditions for non differentiable, and you use the idea of a sub gradient, not a gradient. So there is an extension of it, but we won't do it in this quarter. In fact, generally, this quarter, the way things are gonna work is this. You can actually reformulate your way – well I'd say this is a good topic. Sort of like the next topic. You can reformulate your way around most – almost all non differentiable problems, right? In fact, it's shocking the extent to which you can.

I mean let me just – let's look at an example, right? So suppose someone says, well, I want to minimize – note the period – A X minus B infinity. Like that. That is horribly non differentiable. I mean the objective here is piecewise linear. And, you know, whenever, you know, one entry of A X minus B hits the maximum and takes over the maximum, you've got, like, kinks in the objective. This is horribly non differentiable. Okay?

So you say, well, that's hit. KKT, I can't do it. But interestingly, we know how to – you know how to reformulate that. That was all the stuff we did in the last week or two. What you do is you rewrite this as – I mean there's lots of ways, but this one you might write as T – that's a new variable subject to A X minus B is less than T one, and it's less than minus T one, like that. Okay? Now what's super cool about this problem is it now has N plus one variables.

It also has two M linear inequalities. But notice now – totally differentiable. So I cannot apply KKT to this because F zero is non differentiable. I can apply KKT to this. So you see what I'm saying? So for us – mostly, for non differentiability, we – our policy generally, pretty much, for this course anyway, is to reformulate our way around it. So however, having said that, there is a generalization of KKT directly to non differentiable cases. So okay.

Now we're gonna do something that's actually very interesting here. And let me just – let me give you the big picture first. So here's the big picture, and at first it seems ridiculous, so let me just show you what it is. So suppose I have a problem that's P, okay? And we know how to form a lag range dual. We'll call that D, okay? So we have a fixed procedure by which we can do this. Okay? So that's – and you know how to do it. I mean it's – you take the original problem, you form a lagrangian, minimize over X, maximize that, call that G, maximize G subject to lambda – you know, that's it. That's this thing.

Okay. All right. Now what we're gonna do is this. I mean this is just really dumb. Here's what we're gonna do. We're gonna take a problem and we're gonna do an incredibly stupid equivalence with it, and I'm gonna create something called P tilde – and I mean really dumb. Wait until you see how dumb these – they're nothing – this is a sophisticated reformulation. Okay? We're not gonna do that. We're gonna do, like – you'll see how stupid that these reformulations are, and we're talking really simple here.

Okay? Like introducing a new variable and saying it's equal to the old one. I mean it literally – things like that. I mean that's how dumb it is. Okay? So we're gonna do that. Well – so this is completely trivial, and then of course, well, we know we can form a lag range dual like this. Okay? Now aesthetics would suggest that these two problems should be trivially related. That's what it would suggest, right?

And in a sense they are because, for example, D and D tilde are related because they are the duals of primal problems that are trivially obtained one from the other. Okay? And so you might – you want to add this in. I guess when you do this in math it's called a communitive diagram because it means you kind of – either way you go you kind of get the same thing. You want to do this. Well guess what, this doesn't work at all. There's – these can be – they can look wildly different.

I don't want to say they are unrelated. They are related. They're related this way – only this way. But there's no obvious connection between the two, and we'll see stunning cases where the most mind-numbingly stupid transformations of a problem – if instead of forming a lag range dual you first transform it in a profoundly stupid and simple way, and then form a lag range dual, you get something totally different. I mean totally different. We'll see examples here. Everybody see the picture?

Now at first this looks bad because, well, first of all, it's unaesthetic. You want it a beautiful picture with – you want to – well if you're trained in math, you want to draw that last thing. That's called a communitive diagram. Anyway, you'd want it to work that way. And at first, it's upsetting that it's not there. Actually, it's very, very good because what it – actually, it's gonna be interesting because it means – it means that – so there's the idea of weak sort of – not weak, but there's sort of the – you can do the basic lag range dual.

And that's where you just follow the book. Somebody gives you a problem, you form a lagrangian, form dual function, maximize dual function subject to lambda positive. That's, like, by the book, okay? What – this fact that there's not a communitive diagram here says there's actually some art and – I mean there's all sorts of interesting things. It means that if you call – if you're willing to call this – this one you would really call the dual because it's created by just turning the crank – no creativity.

This one – I guess probably the technical term is it's a dual of that problem. Okay? So generally when people say, you know, derive a dual or something like that, it's actually quite interesting now because it allows you to do some transformation – this is – I'm talking about how this is interpreted on the street. Okay? So if you're in a courtroom and someone says find the lag range dual, you do it by the book, okay? But if you're among friends and things like that, and someone wants to find a lag range dual, you can actually transform it slightly, then form a lag range dual, and then someone would say is that the dual? You'd say, well, no, it's a dual.

And in fact, if you look in the literature, that's why you'll often see these duals with different names. So if you read a lot of the literature, which I don't really recommend, but if you were to do that, you would find things like, you know, the Frank Wolf dual, or the, you know, the Smith dual, I mean – you have all sorts of names, and they just follow this thing. Okay. So, all right.

So I've made these points here that equivalent formulation problem can lead to very different duals, and let's look at some examples of it. I mean – so here's the most – here's a dramatic example. Okay. It goes like this. Ready? A non-constrained problem – actually, you know, we haven't had a discussion of the dual of an unconstrained problem, so let's have it now. It's gonna be a very short discussion by the way.

So there are no constraints, so what's the lagrangian? Well, for each constraint, we should add a term, lambda I F I, to this objective, and for each equality constraint we should add nu I H I. But there are no constraints, so you're looking at the lagrangian. That's the lagrangian. And there are no dual variables. Well there's no constraints. All right. Fine. You've turned the crank. The next step is to calculate G, the dual function. How do you calculate – well you minimize the lagrangian.

So you minimize this. The minimum of that is P star, right? I mean – period. Therefore, there's a dual function, you know, it's a bit – I mean, I don't know, you could argue about semantics here because it's a strange function in the sense that it actually has no arguments. But imagining you could pass something to G, which you can't, it would return the answer P star. Okay? So now – let me tell you the good news.

Good news is, well, there's zero duality gap. By the way, P star is an excellent bound on P star – lower bound. As lower bounds go, they don't come better. But it's completely useless [inaudible]. It's utterly useless. All right. Now watch this. I mean look at this problem in this problem. I mean you can't get a stupider transformation than that. This basically says I'm gonna call – introduce a new variable. I'm gonna call it A X plus B. That's all.

I'm gonna call – sorry – it is A X plus B, I'm gonna call it Y. And so I'm gonna minimize F zero of Y, subject to Y equals A X plus B, that's what this is. Now you have to admit, as transformations go – and by the way, this is so stupid that anybody who starts this way, you should think, look, this is so dumb and simple, there's no good – nothing interesting is gonna come of renaming something. I mean that's all this is. It's just renaming something. Technically it's not.

You've actually introduced a new variable and new equality constraints. Okay? The point is this is no deep, you know, this is not a complicated transformation. Okay. Now the dual here however is this. You form a lagrangian and it's F zero plus nu transpose times A X plus B minus Y, okay? That's – this is the lagrangian, and you minimize that, and you get something interesting. You get minus F zero star plus B transposed nu if they transposed nu as zero. And so the dual is this. This – so here the original problem – unconstrained problem had a completely useless dual.

There was – the dual function was just P star. Just a constant. Now you get something that's not remotely obvious. It's this. It's different. And let's just look at an example. I mean here's an example. Minimize norm A X minus B. Well we do that over the time. This is a general norm by the way. It's not too norm. Okay? So just general norm. Minimize norm A X minus B. Well if you just form the lag range dual – if you're on the stand under oath, and you're asked to form a lag range dual, you say, yes, I've done it.

They say, can you tell us what the dual function is? You say no problem. It's P star. They say, what's P star? And you go, it's the minimum value of A X minus B norm. And then there's nothing they – that's not – that's it. That's all you say. You just say that. They'd say that's the lower bound on the optimal value of this problem, which is also P star, so everything's cool. If you apply this procedure, you end up with this. I mean you minimize norm Y subject to Y equals A X minus B, well, I mean, come on.

Transformations don't get simpler than this. You derive the dual of this. You get something very interesting. You end up with the following. You work out what the conjugate function of the norm is, and the conjugate function of the norm is the indicator function of the dual norm unit ball. I think we've actually done that. But what happens is you end up with the following problem. The minute the dual of this – now by the way, this is sort of a lag range dual, so you would – it's not – you can say the lag range dual exists in this – this the lag range dual of that.

Is this – you maximize B transpose nu, [inaudible] A transpose nu equals zero, and nu in the dual norm is less than one in – well in dual mode, sorry. Okay? So this one is useful. It has huge number of uses actually. Tons and tons of uses. It is not obvious at all. Now by the way, the way this would work is this. Some people would – so there's many ways you could say this. You could certainly say this is a dual of that problem, and someone says really?

And you'd say, okay, look, technically, here's what it is. It is the lag range dual of a trivial reformulation of this problem. That would be if they ask you really – because there's actually other ones, and you'd get other duals even for the norm approximation problem, and that's why if you look in the literature you'll see all sorts of different duals, and you'd see the Smith dual, and the Frank Wolf dual, and then this the dual, and blah, blah, blah like that. Okay?

That's why it is – you could say – you could look up QP duality and find four different duals. Okay? One will be the dual that just turns the crank that we have. Others will be ones where there's some slight reformulation, and then you form a lag range dual. And they'll look different. They'll look as different as these. So anyway, yeah?

Student:[Inaudible] create [inaudible].

Instructor (Stephen Boyd):Nothing.

Student: Nothing?

Instructor (Stephen Boyd):Nothing at all. So here – in fact, let's talk about duality gap. Let's – for this problem specifically, somebody tell me what is the optimal duality gap for this problem? Why? It's actually zero. So in other words, there is no issue. This problem and this problem, same optimal value. Why?

Student: Convex [inaudible]?

Instructor (Stephen Boyd):Yeah, so if Slater holds trivially, Slater says any inequality constraints – for any inequality constraint – so sorry – for the inequality constraints, there must be a point that satisfies them strictly. Well look at that problem. There are no inequality constraints, therefore Slater holds. There's nothing to show. Done. So P star equals D star. Now since these two obviously have the same optimal value, therefore the optimal value of this one and this one hold. So it works.

Okay. So there's another trick, which is this – and we've already seen this a couple of times. You can actually – there's a difference explicit and implicit constraints. It doesn't really matter generally. I mean it's kind of just a trick in labeling, you know, in general. When you form a dual, it is not a trick. So you will get different duals. So if you include, you know, when you have a constraint, you can either explicitly declare it in your list of constraints, or you can sort of secretly attach it to the function, the objective object, as part of the domain. Right?

Everybody – so, I mean, if you look at these originally, these are just sort of reformulations, and it's just kind of trickery. But you can imagine sort of, you know, if you imagine, say, software that implement – you can imagine how this might work. They'd actually be a little bit different. So in one case it's in the domain of the objective – for that matter, you could sneak it into the domain of one of the constraint function or whatever, or you can just have a domain of the problem, but that doesn't matter. So let's see how that works

So here's a problem which says minimize a linear function subject to equality constraints and a box constraint on X. Okay? That's obviously an L P here. You can write it out as an L P. This is an L P. That's the dual L P. And you have lag range multipliers for the two inequalities, you know, that lambda one is for this one, lambda two is for this one. Oh – say, this is fun. Let me just mention something since we've just covered this. It's actually quite interesting.

What can you say about the entries of lambda one? Very poor choice. Sometimes you'd call this lambda minus and that lambda plus or something, you know, because it's the upper and lower bound. Can the entries of lambda – could the third entry of lambda one and lambda two both be positive? And if the answer's no, explain to me why. [Inaudible] that's fine. So – but finish the argument. I agree. Finish the argument.

Student:[Inaudible] that's one and minus.

Instructor (Stephen Boyd):Yeah. So if a third component of lambda two were positive, by complimentary slackness, X three has got to be slammed up against this inequality, and therefore X three has to be one – X three star, right? If also the third component of lambda one is positive, then it says this inequality, X bigger than minus one, has to be tight, and that would tell you that X three has to be minus one.

Now numbers cannot be minus one and one at the same time. So this is impossible. Everybody follow this? So actually, you will see this as actually a practical use for this. It means that lambda one and lambda two actually share – so actually, people will do this, actually, as a data storage – one trick is actually to store them in the same array, and they'll put a minus sign on lambda to code whether it's at the lower or upper bound.

And so they'll actually call this lambda plus and lambda minus. It's actually – it's a good trick. It actually makes sense. But you will actually see that. You go to Google and get some codes, you will find optimal lag range multipliers for what people call, you know, bound or range constraints, they will be returned as – in some cases as one vector where it's reported as positive if it's the upper bound is tight, and negative if it's the lower bound. That's just an aside.

It just had related to what we've just done, so I thought I'd mention it. Okay. All right. So this is the dual. Everything's fine here. No problem. That's another L P. But now what we're gonna do is this. We're gonna rewrite this problem this way, and then there's no – well, okay, clearly they're equivalent. By the way, they're not the same problem because actually you would get different exceptions thrown at you when you pass in X, which violates the – if you give an X outside the box over here, the objective will happily be reported – evaluated.

The objective functional will record its value and pass it back to you. Okay? A X equals B. Well let's say that's zero and it'll give – it'll come back and give you the [inaudible] token. It'll return the [inaudible] token. Then you'll throw X into the box constraints, and one or more of them will come back and say infeas. Okay? By the way, which means that your objective value is interesting but not relevant. Okay?

Now in this problem, if I throw – if I pass in to this problem and X, which is out of the box, the objective throws an exception at me, and the objective throws like an OOD token back at you, which means out of domain. Okay? So that's the difference. Of course, it makes no difference. I mean it doesn't – but now let's form the lag range dual and it'll make a big difference. Let's try it. So here I simply take – the dual function is I take F zero, as usual, plus nu transpose A X minus B, this, that.

And I have to minimize this over – and I've already simplified a bit here. So really this minus one less than or equal to [inaudible] one, to be honest with you, to be fully honest with you, this thing is actually attached to that object. It's an attribute of that object. You see what I'm saying? Because – anyway, I just pulled it out, okay? But I know how to – in fact – wait a minute. I think you've solved one. Didn't you do this on some homework?

This is one of your trivial L Ps. Minimize C transpose X subject to X in some box. Didn't you – I think you did that. I know I talked to people in office hours about that, so – and I think they were in this class. But maybe you did something like that, or something close to that. Anyway, the solution is very simple. You select X to the plus one of C is negative, and minus one of C is whatever [inaudible]. Anyway, if you do that, then the optimal value of this thing is this, it's the one norm of this.

And so you actually just get – sorry. It's not just C. It's A transpose nu plus C like this, and you get that. And that's the dual problem, okay? So here, the originally problem is – and you might, by the way, to make them look more like duals, you might write this this way. Yeah. There you go. So you could say I want to solve – minimize a linear function so there's a linear equality constraints and an infinity norm constraint.

Then – by the way, you might say, oh, hey, here's the dual. It's unconstrained, and you minimize a linear function minus a one norm. By the way, when you write them out by norms, it gives people that you're getting that good duality feeling because you know that infinity norm and one norm are conjugates. And so they're related by duality. So this sort of makes sense. So you would say it depends how well you know someone and what the social context is, but you – it would be okay to say this is – I guess properly you'd say this is a dual of this original problem. Something like that. Question?

Student: How did you get to [inaudible] of X Y?

Instructor (Stephen Boyd):Here?

Student: Yeah.

Instructor (Stephen Boyd):Oh, I collected A transpose nu plus C transpose X, and I worked out this. There you go. Like that. And I know what the answer to this is. I choose X to be plus one if the corresponding entry of A transpose nu plus C is negative. And the optimal value of this thing is the one norm. of A transpose nu plus C. So that's what I did. It's a dual – basic dual norm result. Yeah. Yeah, I'm not expecting you to follow every line here, obviously, I’m going way too fast. I mean you can get most of it, so if you’re getting every line, then I’m going too slow.

So – and I – in that case, I presume I would be getting the international non-verbal signs – boredom signs from you. So I should be getting that from about 10 percent of the class, or something like that. Okay. So that finishes up – I mean that was just a quick topic, but it’s just to point out – and you’ll do examples of this in homework and stuff like that. So – and you’ll see it in – you’ll get these ideas. But the main point was this, and it’s not obvious. The main point is this.

Trivial reformulations of problems, you don’t get a communitive diagram of lag range duality. So just – and it’s interesting. You can get very different duals by doing things that would appear to be silly. I mean even dumber is, like, if you had to take this norm problem, and you minimize the norm, you could say well, look, minimizing the norm is as minimizing the norm squared, you know, obviously because the norm is positive not negative.

If you then form the dual of minimizing norm squared, you get another totally different dual. Totally different. I mean it’s not totally different – it’s got the same problem data in it, it’s also got a one norm in it, but it’ll be different, and they’ll have different uses, different applications, and things like that, so okay. Our last topic – we’re actually gonna skip a very important topic. Sorry, but I’ll tell you what it is. It’s duality for feasibility problems. And these are actually called – I mean, but it’s in the book, we do expect you to read it, and we’re gonna find some homework problems on it I guess, or let’s make sure we do.

At least one or two, or something like that. And in this case, you get things called theorems of the alternative, so I’m not gonna cover it because after you’ve sort of seen a whole bunch of this stuff you can guess it. What I want to do now is – the last topic in duality is problems with generalized inequalities. So here we’re gonna look at – so we have F I of X is a vector. Everything works the same, except remember how it works in our case is we form, you know, lambda I F I in the scaler case, but now F I is a vector. So it’s not surprising that lambda I should be – has to be a vector too, and this becomes [inaudible]. Lambda I transpose F I. That’s fine. Here’s what’s cool though. Lambda I for scalers were bigger than or equal to zero. Now there’s – for scalers, there’s no very many interesting inequalities other than the ordinary inequality, okay? That’s it. So for vectors, you have lots of notions of what it means for lambda I to be positive, and it turns out – and you’ve got – by the way, it had to be for aesthetic reasons. For various reasons, it had to be – lambda I has to be positive in the dual cone. Okay? And if you check carefully, it even sort of makes sense. So here’s what it is. This is the lagrangian. If you look at – so F I has to be less than or equal to zero in the cone K I. If you look at sort of dual cones and everything, you can actually mark each thing as actually sort of being in the primal or dual space.

We don’t generally do that. I mean that would be distinguishing between row vectors and vectors, or, in a mathematical context, linear functionals and vectors. So we’re not doing it in this abstract way, but you can do it that way and it will work out. And lambda I transpose has a row vector. It's a linear functional and it has to be the dual. It can only be the dual cone positivity. So – and everything else works. You have the lower bound property. If these lambdas are bigger in the dual cone – bigger than zero, then G is a lower bound on P star. And actually, everything just – it's the same. I mean these are zero, you know, and then this is a product of elements in the cone – that's in the negative cone, and that's in the dual cone. And inner products of things in the dual cone and the cone are non-negative. Inner products of things in the dual cone and the negative dual cone are non-positive. And so this thing's less than or equal to zero. Everything works. Actually, everything works, including Slater and everything. And I just want to do one example of that. It's a semi-definite program. So you want to minimize C transpose X subject to – by the way, this is called an SDP in inequality form, or something like that. Subject to an LMI – that's a linear matrix is less than or equal to, in matrix sense, another one. And the lag range multiplier here is actually gonna be an element of the – I have an inequality – it's gonna be an element. Also, it's gonna be a matrix inequality. So it's gonna be matrix Z symmetric. And it's gonna have to be non-negative in the dual cone.

Now the dual of the positive semi-definite cone is the positive semi-definite cone, so Z's gonna end up being a positive semi-definite matrix. The inner product between two symmetric matrices is trace – in general, actually, it's just trace A transpose B. But if it's symmetric, it's just trace, you know, A, B, period. So it's the trace of this thing, with a minus sign there, times Z. So that's the lagrangian right there. And you stare at it for a long time, you realize it's affine in X. We know how to minimize an affine function. It's minus infinity unless the linear part vanishes. And so you get this. It's minus infinity unless these things – that's actually that X I component of the linear thing here – unless that is equal to zero, in which case, this all drops away and what's left is the trace Z G. That's this guy. And so this is the dual STP. Okay?

Now by the way, this is not remotely obvious. We didn't use anything fancy here. This is just lag range duality for – it's not immediately clear that these two are intimately connected – these two STPs. By the way, this is an STP in so called equality form. And depending on a person's background, they'll either think of this one as the primal and this one as the dual, or this one is the primal, this is the dual. So I think if you're working optimization, this is the primal and that's the dual. If you work in control or signal processing, that's the primal and that's the dual, but these are just – obviously it's symmetric. Okay. So we will quit here. This actually finishes, essentially, all the base theory for the class. We're done. Now we'll do a lot of applications and stuff like that.

[End of audio]

Duration: 79 minutes