

Instructor (Stephen Boyd): I guess we're on. I guess I'll stay in character and say welcome to 364b and now I'll explain why that's staying in character. I believe we're making history here. This is, as far as I know, the world's first tape behind. Do we know yet if that's correct? Okay. We're gonna have SCPD tell us. I have a thumbs up. So we're making history here. This is a dramatic enactment of the events of the first lecture of 364b. So let me explain a little bit of the background. The class started not televised and we were in the basement of the history building and it was great, I could say anything I wanted. It was fun, but then there was a big broadcast so we had to televise it. So the second, third and fourth lectures were televised and all the rest will be televised, but that first lecture, which was in the basement of the history corner was not televised so we're actually doing a tape behind, which means we're going back and redoing lecture one. By the way, this is just for the benefit since other people were actually here; this is just for the benefit of people watching this on the web so we have a complete set of all lectures. So if you're watching this on the web, you can thank me, but more than that, you can thank the students who actually came to be subjected to a dramatic reenactment of lecture one. Most of them skipped lunch to do it, and yes, there's more than one of them if you're guessing. If you were wondering about that. So 364b, I'll say a little bit about it. Of course, it won't be same as what happened on the first day, but it's just so that we have something there that records the first material. It's a follow along for 364A. I guess what I said on the actual first day was that it won't be televised, I can now revise that to say it is televised and will be televised and those lectures will be available on the web as well. That's how this works.

The requirements for the class are basically homework, one of which is already assigned, in fact, one is already due in real time, but anyway, we'll have homework. Whenever we think of homework, we'll assign them and we'll pipe line them so we'll assign homework before – they won't be sequential so while you're working on homework two, we'll assign homework three and they'll be small and variable amounts and things like that. There's no final exam so no more 24-hour overnight fun. I know you're disappointed. There will be a project and so let me say that's going to be a substantial portion of the course will be a project. I'll say a little bit about the projects and then move on and cover the material. The projects aren't supposed to be a full research paper or something like that. In the past, this course and then several years when I taught the original Convex Optimization course there were few enough people to be having the project. Many of them turned into papers, many turned into thesis and so on. But we have enough people in the class this time that we can't actually do that. The projects can't be 15 page things with long stories and things like that. We just can't possibly read these or help people with them so what we're shooting for in the project now is it's really toned down. You should be thinking the final artifact of the project will be something that's on the order of five pages or something like that. We're going to be very strict on the format and not accepting weird, long and unclear things. It's gonna be in our format period. The idea is you should shoot for something – even if you're interested in something that's quite complex like you do medical imaging or you're in statistics and do something or other, the problems you're interested in might be large, complicated and so and you can't really

do them justice in five pages, that's fine. What you'll do for your project in this class is gonna be something like a characteritute of it, a highly simplified version. One model for that is something like the examples in the Convex Optimization Course so that's your model for a project. They're complete, simple; anybody who knows a little bit of applied math can understand it perfectly. There will be equations there that don't make any sense, but you just have to say, you know, the return is given by blah, blah. You don't have to know that. That might be a whole course to understand that formula, but the point is if you read it, you're just saying there that that's what it is. So that's the idea behind the projects and throughout the quarter, we'll have more to say about the projects. Let me think what else to say in terms of preliminaries.

What we'll do is just cover the material so we have a complete record of the transcript. I should say a little bit about the topics we're gonna do in the class. First of all I should say the class is gonna be more disorganized than 364A; 364A has a very coherent structure, it kind of follows the book and there I don't mind saying to the people who have taken 364A I know we went way fast, there's no way that anybody could actually absorb everything in that book in one quarter. My working hypothesis is that you absorbed 75 percent of it and every now and then I'll take some time to try to remind you of some of it or something like that. And we'll have more time this quarter to absorb some of those topics. So this will be disorganized. We have some lectures prepared. The lectures this time actually have notes associated with them so if you go on the website you'll find slides but you'll also find these more detailed notes that are five pages, 10 – you should read those, we wrote them and it wasn't easy, please read them. Those have more of the details and things like that. It sorts take the place of the book this quarter. So we're gonna talk non-differential optimization, that's the first section of the course so we're gonna cover some radiance, we're gonna start on that today. We'll talk about methods that work for different non-differentiable problems, and I should say a little about that. Of course, in 364A, we dealt with lots of non-differentiable problem, things with L_1 norms, L_∞ norms, and it's just not any problem at all. The way you deal with those in the 364A style is you transform the problem so you have some absolute value, you introduce a variable T and you replace the absolute value with T and then you push on the list of constraints, you know, $x \leq T$ and $x - T \leq 0$ and then now you have a bigger problem, one more variable, two more inequality constraints, but that big problem, you just eliminated one absolute value. And that's our standard method. That's the approach you should use. If you can use an interior point method, by all means do. By the way, that's what CVX does so when you type in a one norm in CVX or anything else, any kind of piece wise linear thing and max them in, it's doing for you this expansion in transformation to a larger, but differentiable problem which is then solved using [inaudible] optimization method. Then in contrast, we're gonna talk about methods that handle non-differentiable problems directly. No transformation. They deal with it, they accept the fact that there's gonna be functions with kinks in them and they just deal with them so that's what we're gonna look at.

They will have other advantages. We're gonna see, later in the class, maybe two weeks in, that they are the key to decentralized optimization methods. We'll see other advantages that they have, but basically if you have the option to solve a problem by

transforming it to a smooth problem and using a cone solver, by all means, that is by far the best method to do it. Okay. So let me talk about some of the other gross topics we're gonna talk about. So we're gonna do sub-gradients, non-differentiable optimization; decentralized optimization, this is really fun, this is something where you would have, in the simplest case, you would have multiple processors or decision makers that are making local decisions and they will coordinate to achieve global optimality. Very cool. Some other topics that we're gonna talk about – and then it gets kind of random and spotty – other big chunks that we're gonna look at our gonna be random ones. Let me think of if I can think of some of those. We're gonna do a bunch of stuff on non-convex optimization. For example, we'll do global optimization, that's where you have a non-convex problem but you're actually getting the exact solution. At the end, you can certify it. Those methods, which you pay for in a global optimization, you pay in is time, so they can and often do run very long. But we'll look at those methods. They're actually quite straightforward so we'll look at those and we'll also look at the other methods where you keep the speed, but what you throw out is the guarantee of global optimality. So we'll look at methods like that. Particularly, we'll look at sequential convex optimization. Other things we're gonna look at will be relaxations a bit for [inaudible] problems, we'll look at some other cool things, there's these things called sums of squares methods, I'm hoping to look at those. Hopefully, we're gonna have a new lecture on model predictive control, but at that point, it degenerates into just a set of cool things that people should know about.

Oh, I forgot one more thing. One more topic is we're gonna talk about scaling convex optimization problems up to way, way big problems. Way big, so the conventional methods – these will scale to something like at 10,000 variables, 100,000, depends on the sparsity pattern. So these will work, but the question is what if something happens and you want to solve a problem with a million variables, 10 million or a 100 million and there's a whole class of methods that basically work on these – they use a lot of the standard and basic methods I might add to scientific computing so we'll look at those. So we actually will have a section of the class on solving huge problems. Here's another lecture I want to add, maybe we won't get to it, but here's what I want to add. I want to add a lecture on solving small and medium sized problems super fast, like, real time applications. I'm talking microsecond's type of things. We know what will go in that lecture, we just need to write it and have it fit in the class. So hopefully, that will work, but who knows. Oh, I should say this. For your projects, you may have to read ahead so if you're doing a problem that involves something that's non-convex, if it's small enough and you want to make a uristic for it, if it's small enough, you might want to go ahead and implement the branch, and the global optimization method and solve instances of it because then you could say something interesting. Okay. I can't think of anything else or remember anything else, but it doesn't matter for the first lecture. What we'll do mainly – and this is the important part just so we have the record is to go on and cover the material in sub-gradients. Okay. We'll start with sub-gradients. Let me just say a little about the idea first. The idea is to directly deal with non-differentiability's so we're gonna generalize the idea of a derivative and a gradients to non-differentiable problem. It's gonna turn out – and this is not uncommon, not unexpected – what's gonna happen is you want to generalize the ideas through calculus to non-differentiable problem, and it's

gonna turn out that there's gonna be two generalizations of the derivative and they're gonna be different. But the cool part is there gonna be related by convex analysis. They're gonna be sort of duals of each other so that's gonna all emerge in this because that's the idea. So we're gonna start by looking at one, which is a generalization of gradient. There's another one. We'll get to that. It's the directional derivative. But we'll start with sub-gradient so let's just start with there. Then we'll cover some gradients and this idea of strong and weak sub gradient calculus. I'll talk about that. And then we'll talk about optimality conditions and we'll see how far we get. Actually, that's lie, we're not gonna see how far – I know how far we get because this is after all, a tape behind. In this case, this is not an initial value problem. It's a final value problem because I know where the lecture is gonna end. Okay. We'll start this way. If you have a differentiable function, you have this inequality. Any convex differentiable function satisfies this. What it basically says is that this is the first order approximation of F at X and it says that this first order Taylor approximation – what calculus tells you is that this thing is really close to that as long as Y is close to X . Really close means close squared is what that says. But what convexity tells you is that this function is a global under estimator of F , and now the questions is what happens if F is not differentiable?

Well, you define a sub-gradient so a vector is a sub-gradient of a function, F , which is not necessarily – and we're gonna do this, this is not necessarily convex. It'll be of most interest when the function is convex, but it doesn't have to be convex. So you say a function is a sub-gradient of F at X if the following holds; if when you form this approximation, it is an under estimator of F globally. Now, we can check a few things out. If Y is equal to X here, if $Y = X$ then you have equality here so basically, this thing, which is an affine thing, it touches its type at the point X , but in a place like this, you actually have some range of options in what you choose for G . You have multiple sub-gradients at a point like that. Now, wherever the function is differentiable, in fact, I'm just arguing intuitively, but it happens to be true – there's actually only sub-gradient. The reason is the tension here, if it rolls any way like this, it'll actually cut into the graph and therefore it's not a global under estimator, therefore it's not a sub-gradient. So in this simple picture here at the point X_1 there is exactly one sub-gradient and it's nothing more than the derivative of this function at that point. It has to be. Over here at this kink point, there's actually two – there's at least two different sub-gradients, but in fact, there's a whole interval of them and it goes like this.

And in fact, the sub-gradient, at this point, is anything in between a little interval and the interval is actually from this lower derivative to the right-hand side derivative, right? The left-hand derivative and the right-hand derivative – we'll get to that later – they both exist and any number in between those two is actually a valid sub-gradient. So that's the picture. So you can put this in terms of epigraphs, you can say that a vector G is a sub-gradient, if and only if, G minus one supports the epigraph. We can say all sorts of other things. Okay. Where do sub-gradients come up? Well, in the next three or four weeks we're going to be looking at algorithms for non-differentiable convex optimization where you just deal with, directly, non-differentiability's. So sub-gradients come up there. And it also comes up in convex analysis so this is sort of the mathematics of convex optimization. Basically anywhere where gradients appear in something – actually, for that

matter, in an algorithm, in anything, we'll see, often, a sub-gradient will pop in and that's it. Oh, if a function is concave, you refer to G as a super gradient of it if the hypo – that's the hypo graph – if this thing lies on top of it like that, and that's actually something like minus a sub-gradient of minus G . Some people, by the way, refer to a vector that satisfies this inequality for a concave function as a sub-gradient. I think that's really confusing but anyway. That's how that works. Okay. So let's do an example. Let's do the max of two differentiable functions. Here's one differentiable function, here's another one and now the question is – so the max looks like this. It goes down here, differentiable, as a kink and then it goes up, differentiable, again. Let's analyze what are possible sub-gradients like at this point right here. At this point, there's only one sub-gradient and it's simply the derivative of this function at that point. Notice that this is totally irrelevant, the second one. Over here, if I say let's analyze sub-gradients here, you go up here and the only thing you get is the slope there. This one is irrelevant. Now, the interesting part is right where the kink point is it turns out that you can get here. At this point, there are multiple sub-gradients and in fact, you can see that – you can actually put something there and you can rock it between the gradient of F_1 and the gradient of F_2 , or in this case, it's in R . So it's the derivative of F_1 , derivative of F_2 . Anything in that interval is a valid sub-gradient and so you get a line segment in this case, an interval in R . That's a subset of R^N and that's the note of this, this differentiable of F , and it turns out that this thing is a closed convex set in general, I mean, for any function, it can be empty. If the function is non-convex, it's clearly gonna have points where it has no sub-gradient at all and that means the sub-differential is empty. So a function, as it curves like this, and you take some point that's sort of in the shadow of whatever is inside the convex hull, there's no sub-gradient there period. Now, that's a closed and convex set. That's easy to show and here are some basic facts. If a function is convex and let's just say if X is away from the boundary – you gotta go eat, right?

Student:

[Inaudible]

Instructor (Stephen Boyd): Oh, no, no, that's fine. That's fine. One of our actresses has left in the reenactment. I should say for people watching this. You don't have to worry, everything here – these are just actors and actresses. So this is just a dramatic reenactment of the events that actually occurred in the basement of the history building 10 days ago. Yeah, so we hired a bunch of actors who look better than the real students, actually. All right. Okay. You have a sub-gradient for a convex function, basically, it points away from the boundary. That's the simple way to say it.

If a function is differentiable, then the sub-differential is the singleton consisting of just the gradient and then it's the other way around. If the sub-differential has only one element in it, then F is differentiable and it's got to be the gradient. The single point is that. So here's an example. The simplest example possible is absolute value so absolute value is very simple. The sub-differential here consists of the set with a single point minus one. Sub-differential over here is single point with the value plus one. Sub-differentials are interesting only at this one kink point, at which point, any number between minus one

and one is a sub-gradient. In fact, I can put something here and I can rotate this from this all the way up to there and I still am a supporting hyper plane, the epigraph and so here it's an interval and when you plot the sub-differential you actually get a beautiful picture. This is actually plotting the set this way and the sub-differential looks like this. It's here – I'm saying if you have a point here, this is one point, and the interesting thing is I've drawn the set this way and I actually get this beautiful increasing line curve. In fact, this is the way it always looks, so if you have a convex function, you're gonna get a nice curve that goes like this and every time you have a vertical segment that corresponds exactly to a kink in the function and the height of that vertical jump is exactly equal to the discontinuity in the slope. Okay. So that's the picture. Okay. So that's the absolute value. Sub-grading calculus. So there's a calculus for sub-gradients and it's very important that you have to distinguish between two and they have different uses and stuff like that. One is – depends what you want to do. A weak sub-gradient calculus is basically it's a formula for finding one sub-gradient of a function at a point. So it's a very different thing. We're gonna find out that some algorithms are gonna require you to just find one. So, basically, you'll have to implement for F , a method called $F \cdot \text{getA sub-gradient}$ and then the argument would be X , and it will return A sub-gradient. It doesn't even have to be deterministic. You can call it twice at the same point and the semantics of that method merely that it returns a valid sub-gradient. It doesn't even have to return the same one. And amazingly, we're gonna find that there are optimization methods that are perfectly happy with this, strangely. So that's what it is. For example for the absolute value it has to return minus one if you're negative, plus one if you're positive; if it's zero, here's what it does. It makes a system call to find out what the time is and then it returns a number between minus one and one that depends on the process ID and the time. That's what it does. Okay. I'm just saying, that's what something like this will be. Now, strong sub grading calculus is actually formulas that can actually calculate the full sub-differential so they return a different data structure.

They would actually return a set of vectors. That's what the sub-differential is. A lot of this is going to be kind of conceptual in the sense that we don't have a way to describe an arbitrary closed convex set. We just don't have such a thing. Except in special cases, this is going to be conceptual. It's not going to be computationally. And here's a claim I make. We'll see this momentarily. I'll back it up a little bit. But roughly speaking, it's this. Strong sub grading calculus can get tricky and there's tons of books on this. By the way, if any of this stuff fascinates you and it is actually really cool, but especially this, the strong sub grading calculus, very cool stuff, tons of books. My claim is something like this – if you can compute a convex function, you can usually compute a sub-gradient, so my argument would go something like this. To say that you can compute a function means that there's some graph, there's some computation graph that you would follow to compute a function, and it might something like a discipline convex programming rule set. There's composition, there's the max of a few things and there's a few rules of affine transformations. My claim is that if I have that computation graph I'm gonna know rules for pushing sub-gradients up that computation graph. So if you make a computation graph that calculates a function, it's very easy to take that [inaudible] and to have it calculate a grade. What you need is for every leaf in that tree has to be able to calculate a sub-gradient of itself. In most cases, you can actually insist the leaves be differentiable.

Now, we're gonna assume that F is convex and I'm not gonna deal with the issues of what happens when X gets close to the boundary and stuff. There's an infinite amount of information awaiting you if this is what you're interested in. So let's looking at some basic rules. The sub-differential of a function is a singleton consisting of a gradient if it's differentiable.

Here's a simple one: scaling. Sub-differential αF is α sub-differential F . Now, here, I am overloading stuff. That's a function, right, which is where the left-hand side therefore has a data structure which is a dated type. It is a convex set of vectors. Sub-differential F is a convex set of vectors and I'm overloading scalar multiplication times a set of vectors in the obvious and standard way so that's set equality here. Okay. By the way, when you see this, you can now imagine in an objective and system. For example, CVX or something, all of this stuff could very trivially be done so basically these are formulas at each computation node or how to get a sub-gradient and you do something, you evaluate a sub-gradient of all your children and then you put them together something. It depends on is that node a sum, is the node a max and so on. And you can get quite far that way. So let's do an example here. Let's do the one norm. So the one norm is actually very cool. It's function that looks like this. It's got four planes coming in, it's got a nice sharp point at zero, but it's got four creases running up the side in the one norm and each one is along the axis where one of the X_i 's is zero in our two. So aligned with the axis with this, you will see a crease in that thing so it's a sharp point at the bottom, four crease points running up at 45 degrees away from the origin. Okay. Now, if you evaluate the sub-differential at any place, randomly, it's probably gonna be differential able there and the derivative is just a sign, a pattern, if X is positive that component is plus one or minus one. It's easy. It will look something like that. If you evaluate this right on the crease – so if you get a crease like this then the sub-gradient is going to be a line segment like this. And if you evaluate at the origin you get this. You get the unit vault in L infinity norm. By the way, this is not an accident. The sub-differential of a norm at the origin is always exactly – and this is just by definition – if the unit ball of the dual norm, so here, the L_1 norm, dual norm is L infinity, the sub-differential is L infinity unit ball. I want to point something out and it's gonna be fun for later. You should think of the sub-differential as the size and this is just a very vague idea, but it'll be made kind of precise later – the size of the sub-differential you should associate with the amount, the non-differentiable of the function at that point.

The kinkiest point is at the origin where it's sharpest so that's kind of the idea and there it's because you can rotate it a lot of directions like this. Okay. So you should be thinking of this this way and this is, like, dual cones, which you remember from 364A, right? When a dual cone is big, fat and blunt, right, it means it's almost like a half space. And what that means is the dual cone is tiny and sharp because you can't wiggle out of a hyper plane too much. So basically, cone big, one, dual cone, sharp. Okay. So the opposite is true, too. If you have a cone, which is super sharp, it comes down to a little fine needle edge. It means when you put a supporting hyper plane down there that's tons of freedom in supporting hyper planes and that means you can have a big sub-gradient, sub-differential. You don't need to know of this. This is just so you have a better picture for what a sub-gradient is. Okay. Now, we get something sophisticated. Yeah?

Student: For [inaudible] you said we choose one of the functions that obtains the maximum?

Instructor (Stephen Boyd): Right.

Student: [Inaudible] sub-gradient methods [inaudible] – what if I choose a function that I notice within two or three percent of the maximum, would that work pretty well or –

Instructor (Stephen Boyd): Yes, it will. Well, there's a name for that. That's called an epsilon sub-gradient and there are methods based on that called epsilon sub-gradient methods and stuff. Yeah. And that's from our friends in Moscow. Oh, I should say something about this. The history of this material is a mathematical topic that's a 100 years old. In the context of actually algorithms to solve this and actually using sub-gradients, it traces back to Moscow and Kiev in the 60s, maybe 50s even. So point wise to [inaudible]. It works like this. Now, I have an arbitrary family of function that I'm taking the supremum of point wise and that gives me my function. Now, here's how you – here, in fact, I'm not gonna give the strong calculus. I'll give you a weak calculus. Weak calculus is really stupid and it goes like this; find a value that actually achieves the maximum and return that sub-differential. Strong calculus basically says find all the ones that do this, take the union of these and take the convex hull. You have to take the closure. In fact, this equation is false in general. You actually need some conditions. For example, the supremum might not be achieved, in which case, you'd have epsilon sub-gradient and things like that flying all around. So if the supremum is not achieved, then this formula is wrong. In particular, the left-hand side is empty and the right-hand side is not so this is where it starts getting tricky.

Student: Why do you need the closure in this case?

Instructor (Stephen Boyd): Let's see. Why do you need the closure? Well, it's not gonna hurt because, as a general fact, the sub-differential of a convex function, at any point, is going to be closed. So it doesn't hurt. But in fact, there's simple examples where, here, if you take the convex hull of all these things, it's open. So you have to do that there. So let's do an example. Is the maximum eigenvalue of a symmetric matrix is an affine function of a variable, so non-differentiable function, we know that. Let me say a little bit about that, the maximum eigenvalue. Here's how you get the maximum eigenvalue. It's complicated. You take the matrix and you form the characteristic polynomial. You do that by writing $\det(S - \lambda I)$ minus A . Now, symbolically, you don't want to see that if A is more than four by four. Trust me. So if A is 10×10 , you can't even see it. It's, like, a long book or more. Okay. You collect the terms and you now have a polynomial of degree n . You might know and might not know that. There's no analytic formula with roots and things like that. Okay. That's a famous result. That, by the way, has no bearing whatsoever. It doesn't stop people from computing roots of polynomials of degree five or anything like that. But the point is, there's no secret formula like the quadratic formula for fifth order and higher. Okay. And then when you calculate the roots, you find the largest root. Okay. All I'm trying to tell you is although, for you, it's not a big deal to talk about $\lambda_{\max}$, it's not a big. I'm just trying to convince you. This is not a simple

function of a matrix. It's very complicated. Now, the fact is that when this eigenvalue here is isolated, so when there's only one eigenvalue at the top, that function is actually analytic. It's differentiable as it could possibly be because it satisfies $\lambda(A) - \max_i \lambda_i(A) = 0$ and actually by the implicit function theorem, that's all analytic.

It turns out $\lambda(A)$ is now an analytic function, locally, an analytic function of A , and therefore of X . So no problem calculating it there. But of course, if there's ties, it gets complicated and you're gonna get kinks and all that kind of stuff so okay. Here's the method. This is nothing but a supremum of this function here is affine in X for each Y so here's how you find a sub-gradient of the maximum eigenvalue. You do this. You form the matrix A of X . You then find its maximum eigenvalue and you find any eigenvector associated with the maximum eigenvalue. That maximum eigenvalue is unique. If it's a unit vector you want, it's two because you could return Q or what do I call it $-Q$, V , it doesn't matter, but Y . So Y . It could be Y or $-Y$ because if Y is an eigenvector so it's $-Y$ and the unit vector. Fortunately, this doesn't change. So then I look at this function here and evaluate it for Y . This thing is affine and the co-efficients are these so that's the gradient and so that's how you get it. It's an eigenvalue value computation. So that's a simple thing. We'll look at a couple of others. Expectation – suppose you have a function of F of X , which is the expected value of a family of convex functions that depend on some random variable U .

Now, this function, by the way, is convex because expectation is gonna preserve convexity. What you need is this. For each possible value, this random variable, you need a selection of a sub-gradient like that, and then it turns out if you take the expected value of this sub-gradient that's gonna be in the sub-differential of X . The way you might do this practically would be something like this and we'll talk about this later in the course. Actually, I can say, if I break character, we talked about it this morning, but I guess I'm not supposed to break character. All right. So the way you would actually evaluate this and practice would be this. What you do is you generate K samples of this random variable. You'd evaluate this and you'd sum over K . Now, if K gets big enough, you'd argue that this, which is actually now, technically, a random variable, its variances are going to zero. It's going to zero, like, $1/K$ is the variance of that. So then you want to get bounds and things like that on this. Here all you do is you pick these, you calculate a sub-gradient and you average the sub-gradient and there's more on that later, which actually was this morning. Minimization. What we're doing is we're just doing the calculus of sub-gradients here. So minimization is this. Recall that if you have this convex problem here, and in fact, to make it simple, we just changed the right-hand sides here. You can think of Y_i as a resource assignment. So you minimize the cost of operating something. You have M resources allocated, by the way, if you allocate too few, this becomes infeasible and this minimum cost is plus. The point is that it's very easy to show, it's standard, that this function – the optimal cost is a convex function of the resources of the Y s. Okay.

So the question is how do you calculate a sub-gradient? Okay. Well, the way to do it is this. It comes straight from something we've already done. You solve this problem and you find an X^* , but also find, and this is what you need, an optimal dual variable

λ^* associated with these inequalities so you take on optimal dual variables. We have a basic formula that says this. The G of Z is bigger than G of Y minus this, that's a basic global inequality from duality and what that says if you turn it around, is it says that minus λ^* is a sub-gradient of G at Y . This is a weak rule. It looks like this. I want to find a sub-gradient of F , which is a big composition function here, H is convex and non-decreasing and these are all convex. Here's what I do. I find a sub-gradient of the parent function, H , the calling function, and that's evaluated at point. Okay. And then I find a sub-gradient of each of the children, the argument functions, at X . Then it turns out I simply form this linear combination here and that's a sub-differential. That's in the sub-differential of F of X . It's a valid sub-gradient. By the way, this formula is the same. It reduces exactly to the standard formula for differentiable H and G for the chain rule. I don't think I will go through the proof here, but you have to argue each step here and so on. It's not that hard to do and you will have to use the non-decreasing argument here and the fact that H is convex. If F convex does have to come in; otherwise, F itself is not convex and so on. Okay. Let's look at sub-gradients and sub-level sets. If you have a sub-gradient at a point so here it is. Here is a level set for a convex function. That's this curve here and this is the sub level set inside here. At this point, it's differentiable, and therefore, the only sub-gradient is the gradient and the gradient points in the direction of maximum local increase of the function and our basic convex and equality tells us something very interesting. It says, basically, any point out here has a function value larger than at F point. That's that basic inequality. F of Y is bigger than F of X plus [inaudible] F of X transposed Y minus X . Right. That's what this says. By the way, that's very interesting. If you were trying to minimize this function, it actually tells you something very interesting and I want you to keep it in your mind for later conceptual use.

So here it is. Let's see. If I wanted to minimize X , and that's my trial point, and I evaluate the gradient like this, then this entire half space I can now rule out. I don't even have to look there because without even checking, every point here has a function value larger than there. And therefore, my search can be confined to that half space. So, roughly speaking, you should have the following idea in your head. If you evaluate a sub-gradient of a convex function at a point, a vector will come back and it'll point in that direction. Basically, what it says then is if you make a hyper plane with this being the normal, so this is the normal to the hyper plane, it says that half space, you do not ever have to look in again, ever. If you want to minimize that function because every point there has a function value bigger than or equal at your current point so I cannot be better. It says, basically, you can concentrate your search on the other half space. And now, I'm stretching things very far here. Very far in my information theory that my colleagues would not approve. I'm going to say this. You evaluate a sub-gradient – very roughly speaking, you get one bit of information about where the solution lies. You're smiling. I know, it's actually not – don't push this too far because it's totally wrong. And my argument goes something like this. If you know nothing and you evaluate a sub-gradient, you get a half space. You've divided the volume where it is, like, roughly, in half. It's a very important thing to know though if you ever wanted to know what's the value. If you evaluate a gradient or a sub-gradient of a convex function, you just learned some information. Okay. Back to the sub-level set. In the case of a point where it's non-

differentiable, there are multiple sub-gradients and every single one of them provides the same thing. It provides you a valid half space where the function value is bigger than there. Something like that. So that's the picture for a sub-gradient and level sets. This idea is gonna come up later in the class and we'll say more about it. So you get supporting hyper planes to the sub-level set. I'll cover this briefly. Otherwise, we're violating the causality and then we'll quit, so I'll say a little bit about this. For a quasi-convex function you have the idea of a quasi-gradient and a quasi-gradient that looks like this. It says, if you transpose Y minus X is bigger or equal to zero and F of X is bigger, F of Y is bigger than F of X .

That's certainly true for a sub-gradient and it simply generalizes the idea so a quasi-gradient, again, the correct idea is something like this. If you evaluate a quasi-gradient, you actually eliminate from consideration an entire half space. That's sort of the idea. I should also issue a small warning here. Warning, invitation, whatever you like. If you go to Google, Wikipedia, whatever and you start typing in things like sub-gradient, quasi-gradients; you'll find an entire industry has been built up in the last 40 years, tons of papers, lots of different things. A pseudo gradient, if its non-zero, is the same as a quasi-gradient and if it's a sub-gradient, then it's a quasi-gradient, but not necessarily a pseudo gradient and you'll see all these kinds of things. Just a little bit of a warning if you do type these things in. You'll find things that are very complicated. So we're only gonna look at two ideas; sub-gradient and quasi-gradient. Okay. Now, at this point, I have caught up to where, in fact, we started lecture two so if I continue, I will violate some terrible law of causality and I hope that going back to the past and re-doing this, I haven't changed anything in the future or anything like that. Things are cool. I want to thank the, more than handful, of people who actually skipped lunch to come and make history at, what we believe is the world's first tape behind lecture. Now, there's always been times when I've wanted to tape lectures behind, but that's another story. Okay. Thanks for coming and we'll quit here.

[End of Audio]

Duration: 62 minutes