Convex Optimization

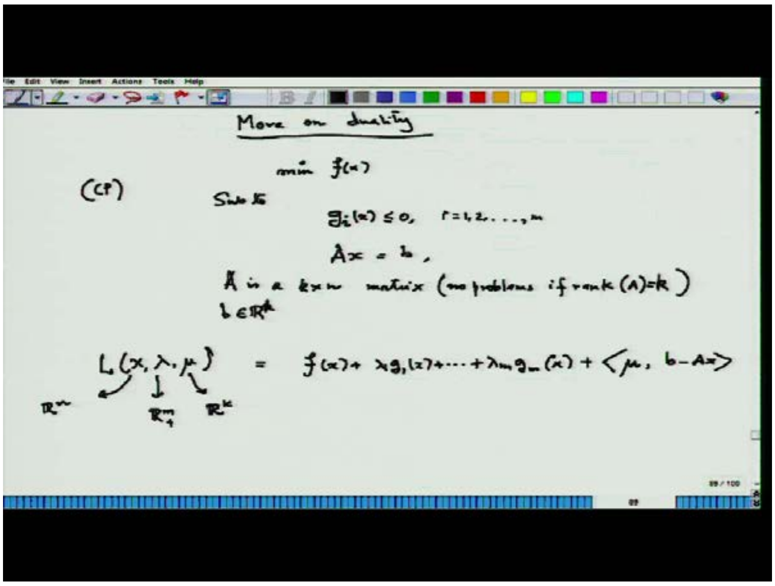
Prof. Joydeep Dutta
Department of Mathematics and Statistics
Indian Institute of Technology, Kanpur

Lecture No. # 17

Yesterday we learnt a very important thing in about optimization that. Whenever there is a minimization problem, there is always an associated maximization problem goes on. This is a sort of duality duality min and max, which is persistent feature of optimization. And we had learned that even this presence of maximization problem at the back of minimization of a minimization problem is quite naturally when when talk about 0 percent, 2 percent, 0 sum games.

So, yesterday we had also learned the important fact that by solving the dual problem we can a priory give a lower bound to the solution of the primal problem. So, thus the dual problem handling the maximization problem in the context of a minimization one makes a pretty good sense, and yesterday we had spoken about something called weak duality. So, today let us you know do something more, we had only spoken about duality in the context of inequality constraints like if you remember yesterday's talk. So, now let us go and talk in the context of equality constants also that would allow us to construct duals or linear programming problems, semi-definite programming problems and…

(Refer Slide Time: 01:52)



Let us also keep on note today's talk would be essentially more on duality. And of course, if time permits today I will try to prove down the duality theorem - strong duality theorem for the convex programming case, but it is not necessary that it will be finished today, because as couple of things involved. So, let us first look at the more general convex optimization problem, CP in the more general form where you are asked to minimize the convex function f subject to and A of x is equal to b where A is a m cross n matrix. Never mind you can even consider no problem, no probes and no problems if ran k is m, this is m sorry m cross n matrix that is full ran, you may or may not, but is in general it is this condition holds. If A is an m cross n matrix, no problem this and b is in R n. So, this quite a general constraint qualification, I would let you as a homework if I had x element of x.

So, how do I construct the Lagrangian this case? That is the thing. Now, how do I express this? I can express Lagrangian now would have x lambda - the Lagrangian multiplier vector associated with this and the vector associated with this sorry sorry I because I have taken here this to be m, I would just change this to make it look much more authentic k cross n and this be k. (No audio from 04:08 to 04:22) Actually when you write on black boards this must like this simple.

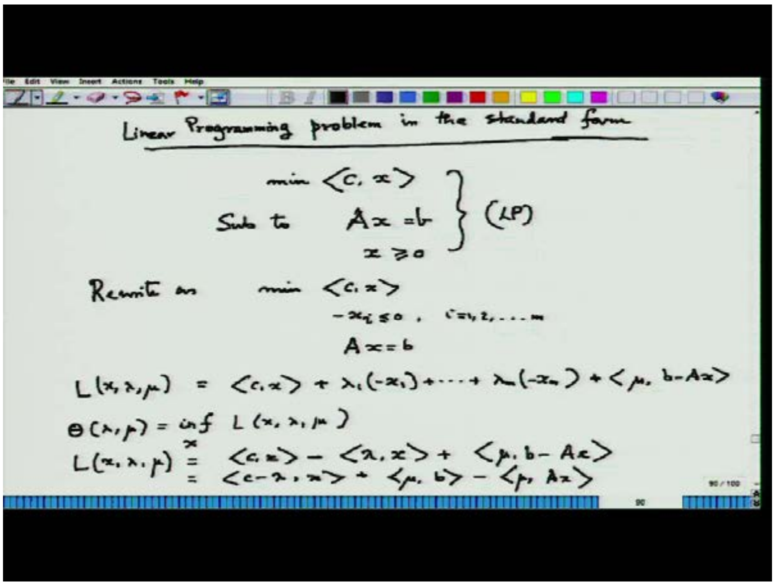
So, now you have to have a mu, remember that associated with the equality constraint the mu has no sign. So, x comes from R n lambda comes from R m plus which consist of

Lagrangian multipliers are inequality constraints and this is a Lagrangian multiplier for the equality constraints, it comes from R k. So, it is usually written as f(x), even place y and z also, I am just taking you might think ok come on guy you have taken y here does not matter y. This is just a change of notation, because we are in the more general set up. So, here…

(No audio from 05:21 to 05:37)

Even also at a x minus b does not matter whichever way it, it is a equality constants. So, this is my associated Lagrangian in this case. So, now this setup defines quite a good class of problems, so how do I compute the Lagrangian or the Lagrangian dual of several type of optimization problem that would be a first goal today.

(Refer Slide Time: 06:07)

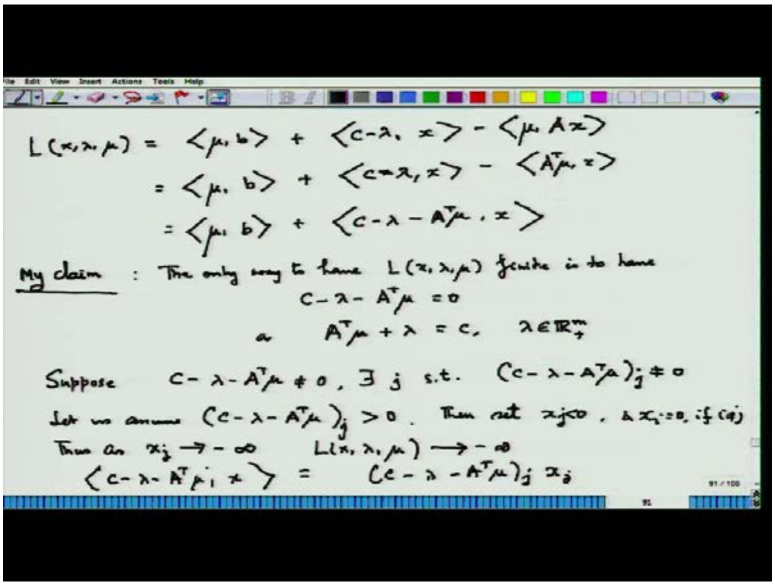


So, let us consider the linear programming problem in the standard form, because very soon we will indulge ourselves in the pleasures of linear programming; linear programming problem in the standard form. So, again I would like to repeat our goal here. The goal is twofold, provide examples of how to construct Lagrangian duals of two important class of convex optimization problems; one is the linear programming problem, another is the semi-definite programming problem. Now, let us look at the linear programming problem in the standard form. (No audio from 07:04 to 07:16) See if I want to write it more explicitly like the form that I have written for CP. So, I should writing in this form. A is again the k cross n matrix of full ran whatever; so, I can rewrite as… So, this is my f x, c of x sorry minus x i less than equal to 0. These are the inequality constraints and this is the equality constraints.

So, once I know this little fact, now I would not like to immediate and write down the Lagrangian by putting the specific f and g i's. So, L(x,lambda,mu) is this let me see in c of x plus lambda 1 minus x 1 plus lambda m minus x m plus mu. So, you will see it is quite simple to do the job. Now, once I know the Lagrangian how do I write down the Lagrangian dual in this particular general case; that would be the second step. So, my first would be to construct a function theta, now would be of two vector variables lambda and mu which is again the same thing infimum over all x in R n of L x lambda and mu. Now, the dual problem for this case, max of theta lambda mu where lambda is element of R m plus and mu is element of R k. So, these are the constraints. So, this is my dual problem; now I it is slightly complicated.

So, now if I want to write down the dual problem for this standard linear programming problem, this is often called LP or LPP - linear programming problem in the standard form. Now, I want to construct this function. (No audio from 10:13 to 10:26) Now, in this particular case, in the context of a linear programming problem what would be theta lambda mu, does it have a specific form; that is the thing that we would like to figure out. Now, let us look at it very carefully and see what is there in. So, if I look at it very carefully, let me observe one thing that L(x,lambda,mu) is written as… (No audio from 11:07 to 11:21) Now, I will club the x things together, so I can write this as c minus lambda into x plus mu b minus mu times A x, this will be as follows, then again mu b minus mu a x.

(Refer Slide Time: 12:00)



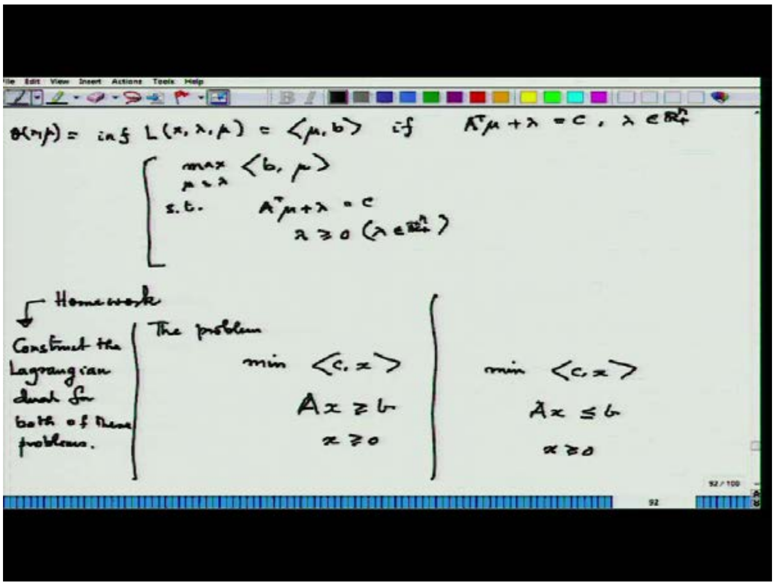
So, see I am gradually coming towards the neater form. (No audio from 12:26 to 12:34) Now, this by simple laws of linear algebra can be written as or of just transpose. So, I am able to write this as… Now, I claim that I am making a following claim. Let us see how can I… The only way… See if I want to minimize this function over whole x it has to be finite at least. The only way to have L(x,lambda,mu) finite right, where is the minimum over all x, it would not be a much important. Unless only we have… See, first of all I need to have this finite, in order to have a descent or a proper dual objective function. So, is to have or I will just remind that this actually means I just inadvertently did not do it, because it is so common, this form right. So, this is my claim.

So, let me see how good is this claim; is it a correct claim or a wrong claim, I have no idea. Now, suppose this is not 0. (No audio from 14:40 to 14:50) So, there must be at least one component which is not 0. So, there exists j such that c minus lambda minus A transpose mu, the j th component is non-zero. So, let as assumed it is done without loss of generality, you could have assume it to be negative and give a similar sort of argument. Suppose this is strictly bigger than 0. Then then set x j strictly bigger than 0 and x i equal to 0 if i is not j then what I can do is; I can keep on increasing the value of x j, keep on increasing the value of x j make it so big and big and big and big. That this function just keeps on blasting off and go towards infinity. Thus as x j plus infinity, because this is when if it is negative, it will be just suppressed one, this becomes very large. So, it blows up. So, it is not finite at all.

So, which means that if I want to minimize, I can show that I can move along one line and sorry sorry and this is one bounded both ways. So, as x j tends to… I should have minus infinity sorry x j tends to minus infinity this thing also tends to minus infinity. See this thing also tends to minus infinity, I am writing I made a mistake. Because now this if you take the inner product then what you will have is that c minus lambda minus A transpose mu, if all the x js other than all the is x is are 0 other than x j. This will only lead to the the value is this c minus lambda minus A transpose mu j, x j.

Now, suppose x j is negative and this is positive, then I can keep on, but this will be negative, so I can go make x j down, down, down, down, down as much as I like and so this whole thing would go towards minus infinity, so this infimum will not have a finite value. So, which means if the similar argument can be set if this is strictly less than 0 then then you can put this to strictly bigger than 0 and go ahead. So, it shows that if I move now for this particular class of x js, if I keep on I am I am generating a sequence which is along which the function value goes down to minus infinity - this function value. So, this cannot have a finite minimum. So, there cannot be a finite sort of dual gap function dual function. So, but if I put this to be 0, it is immediately finite, it is mu b, so if you take infimum over whole x, all the x in R n that answer would be mu b.

(Refer Slide Time: 18:45)

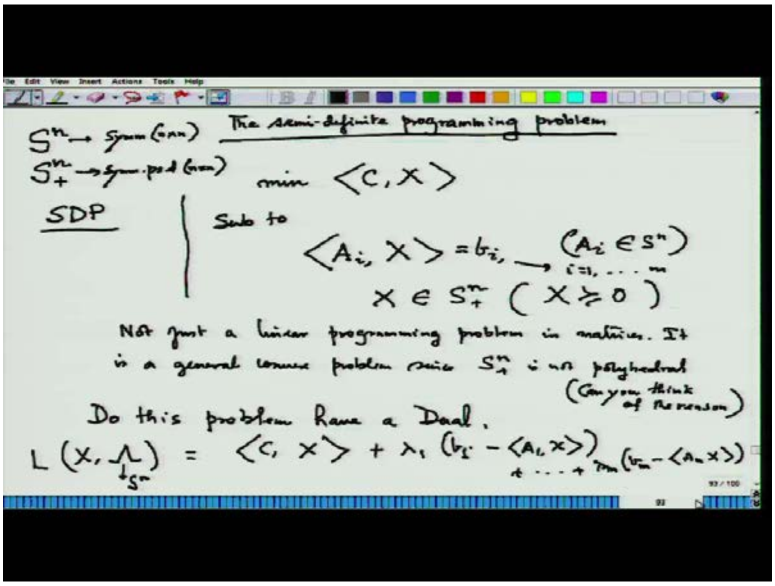


So, inf of L(x,lambda,mu) which is theta lambda mu is equal to mu b. If a transpose mu plus lambda is equal to c and lambda is in R m plus. Assuming that m cross n matrix, I am just writing R m plus v without telling you anything, so you might just get angry of what is this in our case, let take this to be m cross n matrix right. So, we have done a calculation. So now, what is the dual problem? Then I have to dual problem is to minimize b mu over mu and lambda such that sorry not minimize maximize; the dual problem maximize mu and lambda such that a transpose mu plus lambda is equal to c, and lambda s greater than equal to 0; that is in R n plus is means R n plus. This is some, because I have taken x is in R n plus and lambda are the Lagrangian multipliers or the multipliers associated with x, so that this lambda vector is in R n. So, this will be in R n R n plus which is lambda greater than 0.

Sometimes lambda in the literature you will always see… So, this is in place of mu people are writing y in place of lambda people are writing s that we can do specifically we will go we will adhere more to the linear programming community where we will go on to this special set of things that we will study, which I would like to call the pleasures of linear programming. And we do it, because the sub class of convex programming and it is very, very important.

Now, I will give you a homework, the problem is the linear the linear programming problem minimize c x, A x is bigger than equal to b x greater than equal to 0. Another problem is to minimize c x, A x… Now, A x greater than equal to b here means component wise bigger and component wise lesser. Now, the question is construct the Lagrangian dual for these two. So, homework is follows; construct… (No audio from 22:01 to 22:22) So, you will tell me or I will tell you possibly in the next class what are the answers of these. So, now you go to a more general category of optimization problem. Thus SDP is the semi-definite programming problem.

(Refer Slide Time: 22:42)



(No audio from 22:44 to 23:07)

So, you know our theater is in S n the space of n cross n symmetry matrices and the cone that is useful here is S n plus the cone of positive semi-definite matrices. We have symmetric n cross n and sym psd n cross n matrix. Of course, each A i is in S n and x should be in S n plus R n plus is now replaced with S n plus. So, this we have already discussed earlier an important class of convex optimization problem, semi-definite programming problem or colloquially known SDP. SDP as I would like to stress once again that is the hottest area of current research, not because it is something novel, because your decision variables are no longer vectors but matrices, but it has huge application many problems are of this form.

And further for this class of problems, you can write down a polynomial time algorithm which is a very, very important thing and it is very important to notice again that SDP problems have recently shown a great promise in handling non convex global optimization problems. You can actually consider for example, a polynomial optimization problem which is a very hot problem and then it can be shown as as had been shown by larger very recently in this particular decade, in this not in this not decade I would say previous decade in 2002 I guess that give me a problem which is polynomial optimization problem. I can write down a sequence of semi-definite relaxation of that problem and I can solve the semi-definite relaxation by standard techniques which are now well known including the software. And the sequence finally goes and converges to the actual solution of a of the polynomial optimization problem to some one of the actual solutions.

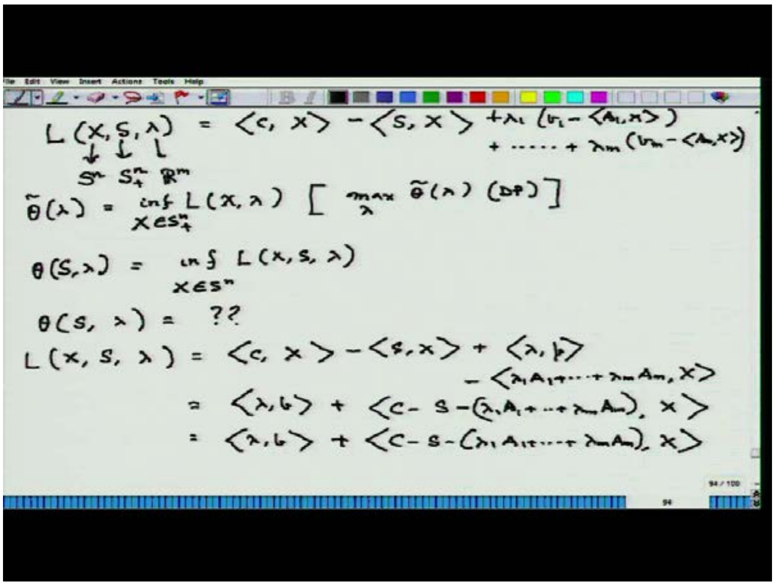
So, this is a very, very big move, because if he got the Lagrange price for this. Because here we are telling that look here is a very difficult non convex optimization problems problem and it is so difficult to solve it, but… Instead of trying to find the crooked algorithm about it, you have a very good approximation which can be whose approximated components can be easily solved, and then you finally can get a quite a very robust in some sense approximate solution.

So, what it means that even when my problem is a non convex polynomial optimization problem, I am actually still in the convex world and that is why convex optimization is such an important area. Now, the question as I told you earlier that this problem is not just a linear problem in matrices. In not it is not just a linear programming problem in matrices; it is a general convex problem. Since of course, those who S n plus is not polyhedral; can you again think of reason why?

Now, question is, does the do this problem have a dual? (No audio from 27:47 to 28:00) I can immediate what I have done for the other case and write the following. I can now construct a Lagrangian. So, how do I construct a Lagrangian associated with the SDP problem? Here I will not use; I cannot explicitly write down the inequality constants that these constants inform of any qualities. So, I will do the following. X lambda is in S n, i equal to 1 to n. So, this x which A i are in S n, so this is also in S n. This can be written as c of x plus lambda 1 b 1 minus A 1 x lambda m b m minus…

Now, once I have constructed this the clever trick is that I am not including this constant which is the hot constant that x has to be positive semi-definite into the this framework into the Lagrangian framework, how to my in the formulation of the Lagrangian, because I cannot write it down in the form of inequalities. I can write down by what is called the Loewner ordering, I can write down like this. But then I at least have no idea how would you bring in that as an inequality constraint here, possibly you can by multiplying with some pd pd matrices. So, but we are not going to handle this way, but we will allow you to think over how to do it; you can do it, but it is not apparent, because you cannot write down the Lagrangian, you cannot write down this in a easy form of inequalities, as you have done for this case; as you have done for the the LP case. Somebody said, let me tell you one thing, we can just possibly extend this Lagrangian a bit.

(Refer Slide Time: 30:46)



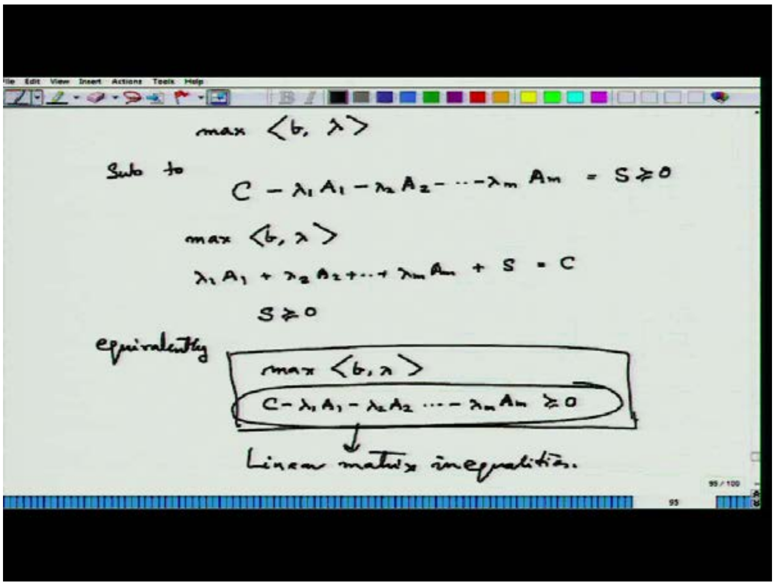
Let me do, I will just write L(x,s,lambda), lambda is a vector right. I am writing this lambda (( )) almost looks like a matrix, but I should write as a lambda. Lambda is a vector, because it is nothing but lambda 1, lambda 2, lambda m; sorry it should not be not in S n, it should be in R m or R n. Now, I can now construct another Lagrangian. I will construct another Lagrangian. This is in S n, this is in S n plus and this is in R m. Let us see, this we will construct like this. Almost an imitation for the linear case plus the remaining same part lambda 1 into…

(No audio from 32:02 to 32:23)

Now, once you know this when again say that; I can either write construct my dual function as theta naught lambda by minimizing l x lambda over x element of S n plus or I can construct theta capital S capital lambda and then of course, maximizing then my dual problem then is to maximize theta hat lambda over lambda. This is my dual problem. And also write this one as not mean I should write in f, but does not matter mean and in things you gradually understand that if the mean if there is no point where the minimum is achieve then of course that is what is the infimum. Here my infimum is not over S n plus, but over S n of L(x,s,lambda).

Now, let me take the second formulation and then try to see how do I compute this, what is this. So, in order to do so I again write down like the I did for the linear programming case. I am rewriting this fact. Now, I have clubbed lambda 1 b 1, lambda m b m together and basically I will have lambda b which we could have written as mu b also minus lambda 1 A 1 plus lambda m A m. I want to again remind those who have forgotten what is the inner product between two symmetric matrices, it is trace of x y. So, I can again write lambda b plus c minus s minus lambda 1 A 1, lambda m A m. See this symbol is called the Loewner's ordering. This simply means that x is positive semi-definite, obviously you do not have to bother too much about the name. Now, I leave it to you to prove that if this expression. So, theta s lambda is finite and is equal to lambda b if and only if this is 0.

(Refer Slide Time: 36:06)



So, by the linear programming type of argument, the dual problem is max of the lambda subject to C. (No audio from 36:17 to 36:31) So, this C minus this… This thing is 0. So, C minus this thing is equal to S and S is in S n plus. So, my dual problem in… In the dual variable I can also write the dual problem as follows. So, as per linear programming this is what happens, of course, I can write it as either if I can write it like this. That is lambda 1 A 1…

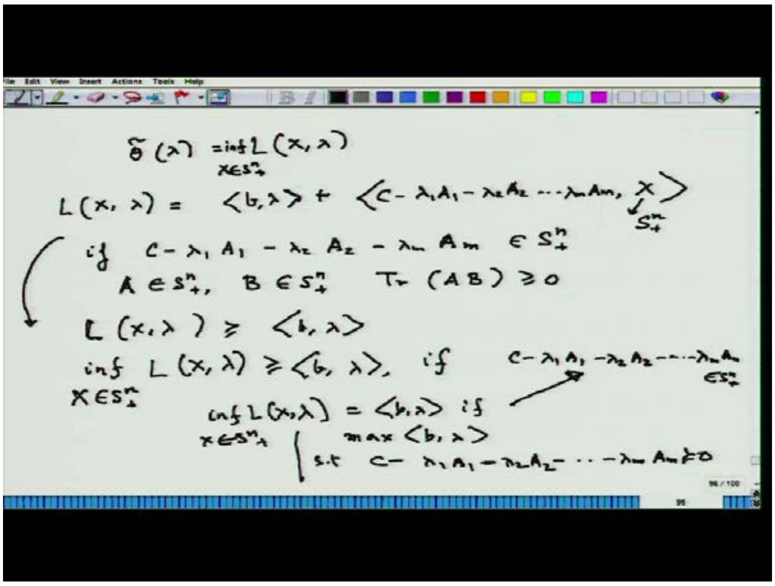
(No audio from 37:05 to 37:19)

Or you can write it as…

(No audio from 37:23 to 37:51)

This last this this this sort of the inequalities or linear matrix inequalities or LMI, as lot of applications in electrical engineering. So, you see we have learned how to construct our duals for both the linear programming case in the standard form and two are kept for homework and the case of semi-definite programming. Suppose I want to take this formulation what would happen; that is the question. So, if I take this formulation that is I have to take in f over x element of S n plus, let us see what would happen.

(Refer Slide Time: 38:38)



(No audio from 38:36 to 38:54)

So, in this case it can be written as b into lambda sorry lambda…

(No audio from 39:03 to 39:25)

Now, you see x is element of S n plus, now you can argue that unless this is equal to S n plus, I cannot say anything; x is in S n plus and if I look into the formulation. Now, x is now in S n plus, I have to minimize over that. So, when do I have - a finite value for this. Now, you see that if this is in S n plus and this is greater than equal to 0 which means if c minus lambda 1 A 1 lambda 2 A 2 lambda m A m is in S n plus, then if you have 2; these are standard result you can figure out yourself. So, if A is in S n plus positive semi-definite and B is also in S n plus and trace of A B is… Of course is not a very standard let us say is a standard result. This is something linked to something was self duality of the cone which is not immediately obvious, but we can we will figure this out in detail and we study semi-definite programming. There is the part of the courses focused on semi-definite programming. So… And we will see how much helpful semi-definite programming is to many, many areas, when it can enter even non convex problem break the bones and non convex problems and give us something.

So, now if this is in S n plus and because x is in S n plus, this would be greater than equal to 0. So, if this is in S n plus, I would have L(x,lambda) will bigger than b of lambda, because this is bigger than equal to 0 right. Now, what is the infimum value? See if I put this equal to 0 then L(x,lambda) is b lambda. So, in fact this is what is true, so I have inf over x element of S n plus if… (No audio from 42:04 to 42:15) So, if this is true which is exactly what we were telling; so, this is bigger than this. But you know at the end I want an inequality, I always want an inequality. How do I get an inequality that means let us ponder.

But, x is in element of S n plus. So, when I put x is equal to 0, the 0 matrix is in also in S n plus this is positive semi-definite. When I put this x is equal to 0 then I get back the value b lambda. So, b lambda is one of the values of L(x,lambda) obtained as I moved x through S n plus. So, which means finally I get inf of L(x,lambda) with x element of S n plus to be b lambda if this holds. So, my dual is again to maximize b lambda such that c minus lambda 1 A 1 minus lambda 2 A 2; I showed that in both ways you can come to the same conclusion. So, then any way you can prove proceed, again I am putting the Loewner ordering.

We have no time to prove the strong duality result today and we will end the topic today here, and in the next lecture we would talk about the proof of the strong duality theorem, and show that if the slater condition does not hold for a convex programming problem. We can give examples for strong duality phase. If a non convex problem duality does not told. See this this story of constructing Lagrangian dual is respective of whether the problem is convex or not. But here since we are concentrated on convex problems. We will show that even for a convex problem, if slater condition fails strong duality goes. We will have examples even for semi-definite programming problems. Thank you very much.