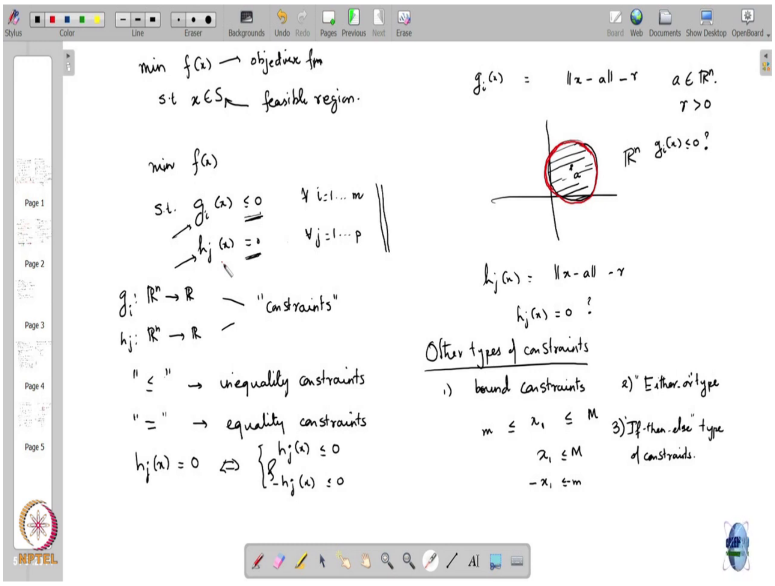
Optimization from Fundamentals
Prof. Ankur Kulkarni
Department of Systems and Control Engineering
Indian Institute of Technology, Bombay

Lecture – 3C
Different Types of Constraints

(Refer Slide Time: 00:22)



Alright. So, let us now let us come back to Optimization problems again. So, as I said we have the way we write optimization problems we write f which is your objective function, and then we can write it as such that x belongs to a set S. So, f is called the objective function, and S is called the feasible region ok.

Now, usually this sort of way of writing an optimization problem is not very conducive for many things for both analysis as well as for computation ok. We, so when you write that x belongs to some set, how do you even specify that set becomes a question? What does it mean that it belongs to the set?

Usually a set is a collection of points, but you cannot list out all the points in the set right, usually there will be infinitely many. So, you have to describe the set in a certain way. So, the way we describe it is that we say we will put a bunch of conditions that x must satisfy. You do not actually list out all the points, but rather only list out the conditions that x must satisfy.

So, those so the way you write it is like this. So, we write as you minimize function f such that g i of x is less than equal to 0 for all i ranging from say 1 to m; and j of x is equal to 0 for all j say ranging from 1 to p. So, now you have defined an optimization problem using some additional functions.

You have of course your objective function, but then you also have all these functions ok. These functions so here just like f these are functions g, g i and h j, they are both they are also functions from R n to R. And they are called these are referred to as constraints. Now, you can see that I have classified the constraints in a written them in a certain specific way, and I have classified them into these two categories.

First is that I have written out separately constraints that have less than equal to one type of requirement, and another type of other constraints which have a equal to type requirement ok. Everything that is less than equal to can be also written as a greater than equal to by just multiplying both sides by minus 1, so that is not a, so I do not need to separately write also a greater than equal to in this ok.

Can, you ok. So, sorry, so these be less than equal to constraints the ones that have a less than that are to be satisfied with the less than with a less than equal to sort of requirement. These sort of constraints are called inequality constraints. And the ones that are to be satisfied with equality, these are called equality constraints.

Now, can someone tell me, can I is it possible for me to reduce everything to my problem further and write it only in terms of inequality constraints? I want to write I want to write my problem only in terms of inequality constraints. So, how do I do that? Right.

So, this is the that is the correct suggestion. So, you can the requirement that h of j of x is equal to 0, this is equivalent actually to two different requirements. One that h j of x should be less than equal to 0, and also the and remember both and h j of x minus h j of x is also less should also be less than equal to 0. So, if this and this hold, then this, so if h j of x is both less than equal to 0 and minus h j of x is also less than equal to 0, then h j of x is should be equal to 0 ok.

So, it is therefore, possible to collapse this problem even further and get rid of also the equality constraints, and write them as two opposing inequality constraint. However, we generally do not tend to do that, and there is a good reason for this. The main reason is that the geometry of the problem is very different for with of an geometry of an inequality constraint is very different from that of a equality constraint ok, and so that is the thing I want to first, I want to sensitize you to ok.

So, let us suppose we are we have suppose a say a constraint like this, say g i of x is suppose equal to negative of x minus some a the whole square. Now, if g i of x is something like this g i is a is this sort of function x minus a the whole square. Then what is the region? And suppose we are so suppose we are taking this from R n to R n.

So, let us write this as g i of x as x minus some norm a norm of x minus a the whole square where a is some vector in R n ok. So, you are fixing some vector a, and then looking at this as a function of x minus negative of norm of x minus a the whole square. So, now, g of g i of x less than equal to 0, I want to look at this region ok. So, I am going to look at this in this is my space R n. This is suppose my point a in that space. So, then what are all the points x; such that g i of x is less than equal to 0, what are all the points?

Student: (Refer Time: 07:50).

All the points right. So, this is this is actually the full space ok. So, ok, so let us change this. So, let us suppose we have I write I can instead of taking a norm of this let us do. So, I have fixed some r greater than 0, I have fixed of a in R n I am looking at the function g i of x which is norm of x minus a the whole square minus r alright. Now, tell me what here is my point a, tell me what sort of region satisfies g i of x less than equal to 0?

Student: (Refer Time: 08:49).

Yeah. So, this is a sphere of radius r centered at a. So, this is the entire region. Now, what if I had if I had a function h j of x, so this norm of x minus a the whole square minus r and I look for the region h j of x equal to 0, what sort of xs are these?

Student: (Refer Time: 09:27).

This is only the shell right. Now, ok, so now, let us can you tell me what is the difference in these two regions? The first region is that is the first both of these sets are actually closed sets ok. Then the shell itself is a closed set, the ball of radius r centered at a with its shell is also a closed set. But the key difference is that the ball of radius r centered at a that also has its interior included in it ok.

Now, whether the interior is when we write an inequality constraints, we inequality constraint, we are what effectively happening is that we are considering both the shell as well as the region inside ok. So, as a consequence the it is implicit in the problem that the that what you have defined is a region.

Whereas, when you are talking of an equality constraint, what you have defined is really a surface. The geometry of these two is very different, and the and this will become clearer as we go further into the course. The main reason is that an optimization problem behaves very differently when the task of optimizing is very different from whether you are in the interior or whether you are on the boundary.

If you are in the interior, you have the entire world you can explore around, because there is a ball around your point which is lying completely inside your feasible region.

So, your algorithm or whatever methodology you have, allows you to explore in every possible direction. While you once you reach a boundary, there are only certain directions in which you can go in, and you have to worry about whether you are violating this or violating that ok.

So, there is a phase change that happens when you go from interior to boundary. When we are right that is why all method is it is always a good idea both for from a computational and analytical standpoint; to know whether your constraint is truly an inequality constraint or truly an equality constraint, and that is the reason why we separate these two out always.

Yes.

Does not matter. It is a this will be a ball of radius, I guess square root of r, my mistake and let me get rid of the 2 ok. Now, there are other types of constraints. So, let me mention this. So, let me, so there are other types of constraints which I have which so for example, you can write constraints that are; for example what can be said are bound constraints. Bound constraints means they just look at a specific variable say x 1, one variable and specify bounds on that variable; saying that this variable cannot exceed a certain value capital M and cannot be greater than a certain other values small m.

When you have bound constraints like this, you can of course think of these as two different inequality constrains, but then bound constraints are often you know the method or the type of solution algorithm you are employing.

It again needs to know if there is a bound constraint usually that is needed and that again will become evident to you, because the fact that there are it can the this will if you do write this as two different inequality constraints, it will look like two different opposing inequalities of the same kind just with a different right hand side ok.

So, it will look to the algorithm as if you have written one inequality like this, and another inequality like this. And this can lead to some problems ok. So, to avoid this sometimes algorithms ask you to specify this separately ok. So, these are called; these are called bound constraints.

For analysis, this usually does not matter. So, we do not we can always observe these as inequality constraints. There are other there is another type of constraints which are constraints that are often either or type so either or type of constraints. So, when I wrote out these constraints these m inequality constraints and p equality constraints all of these remember that we want all of these to be satisfied.

It is not that we have any of these or one of these or whatever, the requirement is that all of these have to be this is a full list of things that you need to satisfy ok. Now, it slightly different type of constraint will arise when I tell you that it satisfy either this or that alright that is a different type of constraint. Now, that also can be modelled in this sort of form, but it takes a lot of effort. We will see if we can teach that later in the course ok, those are either or type constraints.

And even more complicated type of constraint is then you do either or and then take a decision on top of that, so then you so it is if then else type of constraint ok. If you; if something is satisfied then you do this; else if that is not satisfied, then you do that ok, so that would be an if then else type of. So, we will get to these, we will look at these later in the course ok.

So, for the moment, just we will be working with only inequality and equality constraints. So, from an engineering standpoint, we should remember that usually this the way this arise is that you know an equality constraint is a hard is a hard constraint such that this function has to be 0 ok.

So and so value, so and so must be, so and so value is so and so bunch of variables must satisfy a certain hard requirement. This sort of hard requirement usually comes from physical from the physical nature of the problem. It is a, it is a law of nature that for instance that something must be equal to something right.

Whereas, the inequality constraint is usually a human requirement you do not want to x be, you are you have a certain tolerance for you do not want delay to exceed a certain amount, you do not want price to exceed a certain amount, you do not want area to exceed a certain amount etcetera. But you are you do not mind if it is less than that, so that is sort of that. So, these the inequality constraints usually arise out of human specifications ok [FL]. So, I will we will stop here now.