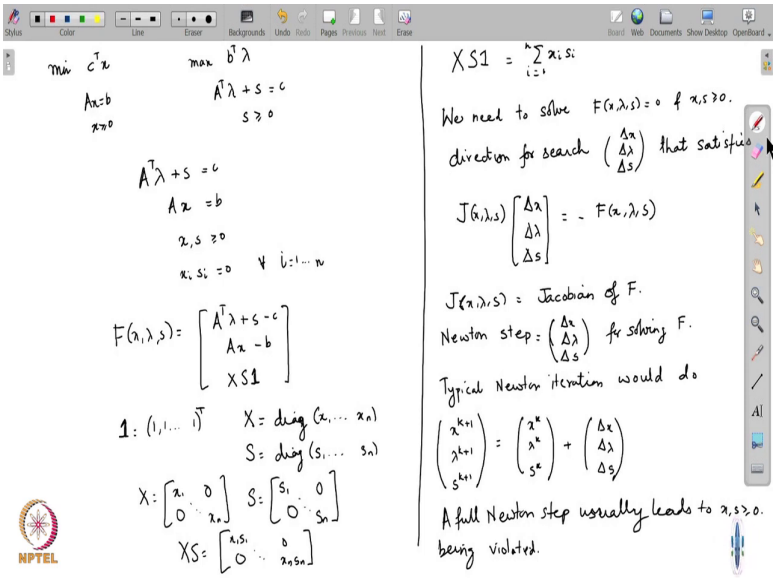
Optimization from Fundamentals
Prof. Ankur Kulkarni
Department of Systems and Control Engineering
Indian Institute of Technology, Bombay

Lecture - 22
Interior point methods for linear programming

Welcome everyone. So, today now what I will talk about is an another type of optimization method that is a full blown primal dual method; that means, it is the method that works in both primal and dual spaces and in and moreover it gives no preference to either space ok.

It works simultaneously in primal and dual spaces as if they are alike as if they are of treats them on equal footing and hence computes the primal and dual optimal solutions together simultaneously. And this method is what is called an Interior Point Method ok.

(Refer Slide Time: 01:01)



So, an. So, the. So, I will show you this interior point method only for the case of linear programming, because that is where it is easiest to explain although this method has been extended now for non-linear programming also ok.

So, what is the idea? The suppose here is my linear program, I am minimizing c transpose x subject to Ax equal to b and x greater than equal to 0. And what is its dual? Let us write its dual in terms of Lagrange multiplier lambda or its dual variable lambda.

So, its dual is this, maximize b transpose lambda subject to A transpose lambda now, A the usual dual is that its A transpose lambda less than equal to c, but instead of writing it this way what I will do is I will insert also, I will insert a slack variable. So, I will write this as A transpose lambda plus s equals c and put s as greater than equal to 0. Now, what we have seen form our from our studies of complementary slackness and so on.

Is that x and s are complementary variables. So, they would end up satisfying complementarity slackness or complementary slackness it means. So, x and s are complementary variables. So, they would end up satisfying complementary slackness which means either x for each component either x i is each component i either x i is 0 or s i is 0 at the optimal solution right. Now, one way of writing this these optimality conditions is as follows.

So, I will. So, optimality conditions would be that A transpose lambda plus S equals c; I must also satisfy Ax equal to b, this ensures primal and dual feasibility, I must also have x comma s both greater than equal to 0 and I should have x i s i as I said for every component either x i is 0 or s i 0. So, this is equal to 0 the product x i s i is equal to 0 for all i from 1 to n right. So, the difficulty in solving a linear program lies precisely in ensuring these two conditions.

Out of these two conditions that have to hold simultaneously is what makes the problem difficult. The first 2 equations that one needs to solve a simply linear equations. So, those can be solved you know in by any classical technique. The trouble is we have to also satisfy x and

s greater than equal to 0 and at the same and also the complementary slackness. Of these even the complementary slackness condition we may say is eventually a non-linear equation.

So, it could be combined or clubbed along with these two as just as something of the same category as the first 2 condition, first 2 equations as the first two equations are simply look linear equations well, this is an additional non-linear equation, but then the trouble is also this particular inequality this is now an inequality that needs to hold in addition to these linear stroke non-linear equations right. So, this is the so, the challenge is ensuring that these two are satisfied and that is what makes this complicated.

So, the way to die so, the we will what we will do is, we will be trying to solve these as non-linear equations and at the same and buts, but our effort at solving the non-linear equations would also ensure, we will solve these as non-linear equations while always ensuring that x and s are greater than equal to 0.

So, effectively we are going to solve equation the first, second and third equation as non-linear as a system of non-linear equations, while ensuring that we do not violate equation 4 alright ok. So, that is the idea. So, now, to in order to express this, let us write denote F x lambda s, denote this consider this function F x lambda s and write this as this function is going to be A transpose lambda plus s minus c Ax minus b XS and. So, this is XS and yeah A transpose lambda plus s minus c Ax minus b XS and 1.

Where 1 remember is my vector of 1's is a column vector of 1's. Now, what is capital X and capital S? Capital X is simply a diagonal matrix comprising of the components of the small x. So, it would you can write this as diag of x or equivalent of diag of x 1 till x n and S as diag of s 1 till s n ok.

So, X is this matrix x 1 till x n and 0s elsewhere and S is this matrix s 1 till s n and 0's elsewhere. So, X times S is simply going X times S is going to be another diagonal matrix. So, XS therefore, is another diagonal matrix where you have x 1 s 1 dot till x n s n and 0s elsewhere.

And that times 1 the vector 1 is XS1 is simply going to give us this XS1 is simply summation of x i s i, i going from 1 till n alright ok. So, this so, what are we then looking for? We are looking for we want to find as so in order to solve the optimization problem, we need to solve F of x this equal to 0 and x comma s greater than equal to 0 alright.

Now at the way to we will do this is, we will do this iteratively we will takes steps at each for at each iteration and we will move in certain direction at each iteration. Then the direction that we want to, but we remember this is not just about solving non-linear equations, this is we also need to maintain feasibility with respect to x and s greater than equal to 0 this.

So, this additional constraint means that, it is not vanilla non-linear equation solving so, but we can take inspiration from the solution of non-linear equation. So, we can write say we can take say for instance a Newton step ok. So, we can take something like a Newton step.

So, for instance, we can look for direction a direction for search as delta x delta lambda delta s that satisfies this. Now, what have I written here? I have written J of x lambda s is simply the Jacobean of F, this is simply the J of x lambda comma s is simply the Jacobean this is the Jacobean of F alright. So, in short what this equation is?

So, this is you can one can see that this is actually nothing but a Newton step. So, delta x delta lambda delta s, simply a Newton step for solving F. Now, if you. So, this. So, in a typical Newton iteration, a typical Newton iteration would do, Newton iteration would do x k plus 1, lambda k plus 1.

And s k plus 1 equal to x k lambda k s k plus the step that we have right which is delta x delta s delta lambda delta. This is would be the typical Newton iteration that it would you start with your original your the point where you are compute the direction and you add simply that direction to the point where you are to get your new to get your new iteration.

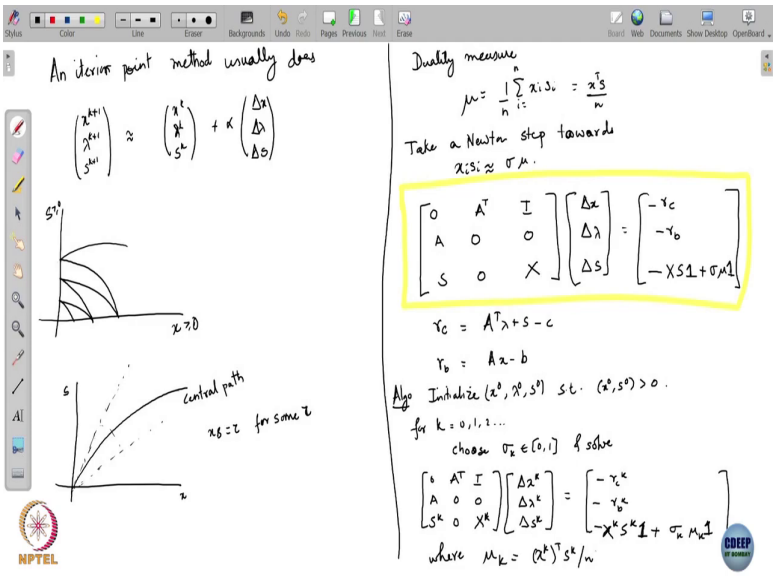
But unfortunately if you just simply so, what this would amount to is taking a full step along the Newton direction right. Newton's along the full taking a full step along the Newton

direction or than along the Newton step. Now, that usually leads to a problem which is we need which is that? We need to remember maintain also feasibility with respect to the inequalities.

So, if one takes a full step along this direction. So, fulls a full Newton step, usually leads to x comma s greater than equal to 0 being violated, full or typical Newton step ok would lead to this being violated. So, what we can what we need to do therefore, is to search along the Newton direction alright, but search to the point where we do not violate this this x comma s greater than equal to 0 right.

So, the idea is to take inspiration from the Newton method, but do not go whole hog into the Newton method because, we are one you are not really solving non-linear equations in the most basic form right. This is a much more specific and structured problem. And we are developing a our own way of solving that particular problem for ok alright.

(Refer Slide Time: 13:56)



So, now if you. So, what are an interior point method? An interior point method usually does x k plus 1 lambda k plus 1, s k plus 1 as approximately x k lambda k s k plus alpha times delta x delta lambda delta s ok. This is what it usually does alright. Now this the so, you now this also can be diluted further.

So, what one even here you know take you do not want to go all the. So, in an it is not merely that you do not want to while you know, it is not merely that you want to maintain feasibility with respect to this. And there is actually this is one of the insights that led to this method becoming so successful.

At one does not really want to its not just that you want to stay feasible with respect to this. If you did that what you would end up doing is you would take a Newton step to the extent that

you can without violating feasibility. So, you keep going along the Newton direction and you will stop when you hit one of these quadrant lines right.

You would when you hit one of these, one of these quadrant half spaces, you would stop. So, you that would then take your iteration in the sort of in this sort of manner, if these were, if this is your if these were your axis say this is the x axis and this is the s axis what it would do is?

It would take your iteration here, then here, then here, then here, then here, then here and so on. Effectively you would try to avoid you would keep taking steps that would go all the way to the point where you will take as large step as you can without violating the x greater than equal to 0 and s greater than equal to requirement. So, this is the x greater than equal to region, greater than equal to 0 region this is the s greater than equal to region right.

So, you would go you would say starting from here, you would go up to the point where s becomes 0 sorry x becomes 0. Then from here you would go when x becomes 0 and so on and so forth. Now, the an actual primal dual method, usually does something even less aggressive it does not, it does not go even this far.

What it would try to do is it would if it would try to look at this kind of a, kind of central region here ok. And effectively it would try, it would make sure that your iterates remain within this central region ok. So, more concretely the terms that is often used is what is called a central path ok, a central path.

A central path is simply a is simply points focus of points that satisfy x comma s x times s equal to tau for some tau ok. So, and as you vary the tau and as you vary and as you make the tau smaller you would you, your you end up getting this central sort of a path.

Now, and what we want your method to do is to follow this particular path that is given to it. Now as a result what it does is, it does not the method does not sort of zigzag or go over from

one form a point where you know one point where s is 0 to a point where x is one of the x is 0 and so on and so forth.

So, effectively it actually does not ever hit any of the corner points of the polyhedral either the primal or the dual polyhedral. And what it actually reaches it reaches to the final actual solution through the interior of the polyhedral; it gets to the eventual corner point solution, but through the interior of the polyhedral right.

So, that is the origin of also the name the interior point method right. If it did exactly any of the early any of these things, then it would be simply another way of doing what any of the older methods for linear programming were doing, which is simply searching over corner points of the linear of the polyhedral.

Since we know the solution lies at a corner point, it would, you would be simply going from one corner point to another looking for the solution. So, the interior point method actually abstains from doing that or refrains from doing that and comes and is therefore, able to move directly to the solution ok. So, and now so, the so, that. So, the actual form or the diluted form is the following. So, what we what in order to define it, let us define this particular thing it is called the duality measure ok.

So, duality measure is mu defined as the 1 by n summation x i s i. So, the duality measure is simply a on equivalently written as x transpose S is divided by n. The duality measure is a measure of how far you are from complement from satisfying complementary slackness.

Complementary slackness would require the duality, if it satisfied the duality measure would be exactly 0 right. So, that is. So, it sort of captures that particular property. Now, what it the what your the actual new interior point method would do is that it would not it would take a Newton step, but move take it to the point where, it to the point where it in towards the point where you are where x i s i is close to some multiple sigma of mu.

So, what we want to do is it does take a Newton step towards this particular path. So, in short it it is taking a Newton step, but trying to make sure that, it is it remains in this in the vicinity

of this central path right. Because x i it wants all the products x i s i to be around approximately sigma times the duality measure. The duality measure is an average of this x i s i's. So, it is trying to make sure that all of them are in a within a multiple sigma or roughly a multiple sigma of the average alright.

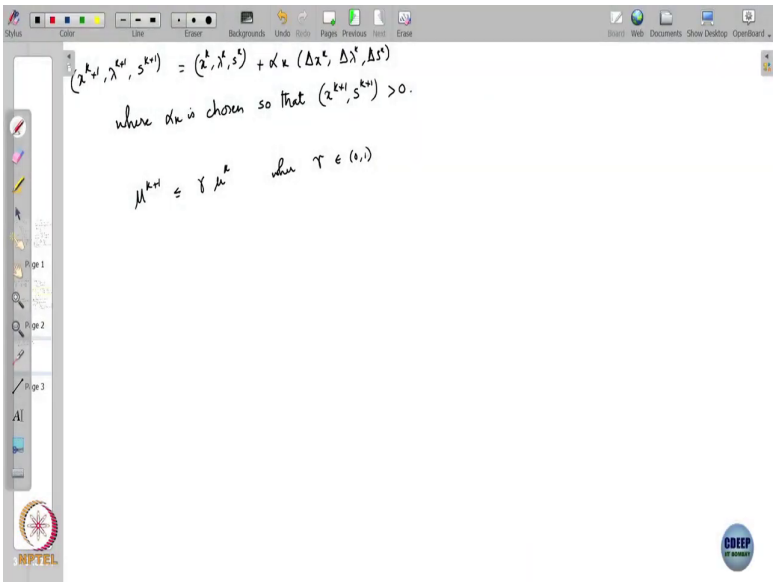
So, then the modified step is then become something like something like this. So, the Jacobean evaluates to A transpose I, A 0 0, S 0 X and then you have your step that you want delta x delta lambda delta s equals; now, the you have the right hand side remember was minus F and minus F would be. So, this is now equal to minus r c this is minus r b and minus XS1 plus sigma mu times 1 ok. So, what is r c and r b? Well, r c is simply the residual ok the residual that remains from the dual constraint.

So, the residual in feasibility of the dual constraint, r b is the residual in feasibility of the primal constraint. So, the actual new the what the actual method is basically do doing is simply this ok. So, the its trying to find its taking steps in such a way that it satisfies this particular equation that is that has been written here ok.

This is the master equation. So, let me write out the entire algorithm. So, the algorithm then is that initialize with initialize your x 0 lambda 0 s 0 such that all the components x 0 and s 0 are greater than 0 ok. Now, for k equal to 0, 1, 2, dot dot dot you solve the following.

Choose a sigma k in 0 to 1 and solve 0 A transpose I, A 0 0, S k 0 X k ok. And you get a delta x k delta lambda k delta s k equals minus r c k minus r b k minus x k s k times 1 plus sigma k mu k times 1. Where mu k is simply the duality measure at step k is equal to x k transpose s k divided by n alright.

(Refer Slide Time: 27:00)



You set x k plus 1 lambda k plus 1 s k plus 1 as x k lambda k s k plus alpha k times delta x k delta lambda k delta s k where alpha k is chosen so that x k plus 1 comma s k plus 1 are greater than 0. So, already what it so, the way this algorithm has been specified it, what it is already doing is it is sort of tilting your search direction in it more towards the central path. And you choose any you choose an alpha k, so, that you can in such a manner that you do not violate feasibly.

So, it is taking a step in the along a modified Newton direction, a Newton direction tilted slightly more towards the central path and asking you to proceed in that direction to the point where you do not violate the feasibility of the inequality constraints alright. So, this is basically the idea. Now, there are much more concrete ways of making sure that you do not

violate feasibility and you move towards the central path. For example, it is possible to define a neighborhood of the central path.

And around that neighborhood and you keep searching till you reach that particular neighborhood that is a more practical way of doing what the of implementing this sort of a method. Those are all details that can be that can always be plugged in, but this is basically the overall the what I have explained is the overall structure of a primal dual method. So, what is the, what is the advantage of what is the, what is the result that we can expect from a primal dual method?

So, primal dual method is effectively what it will what it does is. So, what is that one can expect from the primal dual method? Why what one gets a result that which essentially shows that your duality measure satisfies something like this, is less than equal to some gamma times mu k where gamma ok where gamma is a constant that is between 0 and 1. So, the duality measure and it and the constant also depends it might depend on n, but it is a constant between 0 and 1.

So, the duality measure has this particular property, that it keeps it goes down to 0 geometrically right. So, gamma mu k plus 1 is always less than equal to mu gamma times mu k, where gamma is some constant between 0 and 1. So, which means that for as k becomes larger and larger your duality measure shrinks.

And once your duality measure goes to 0, you would have satisfied complementary slackness. And the way you have designed your search is that it is it has always maintained feasibility with respect to with respect to this particular constraints. So, you have not only satisfied the linear equations, you have also made linear or non-linear equations you have also satisfied feasibility and then eventually satisfied complementary slackness right.

So, that effectively has ensured that the. So, as k goes to it becomes larger you end up satisfying complementary slackness and that effectively ensures that your method has converged to this solution right. So, this is basically the essence of an interior point method,

you what you can see the things that I mentioned to you at the start of the at the start which is that it has made no distinction between the primal variable and the dual variable right.

It has searched simultaneously searched or; however, you want to call it, it has computed simultaneously the primal and dual variables. And its effectiveness lies in being able to do this in that it has computed both for you it has attacked the problem jointly in the primal dual space.

And in that space really optimization is not about a function over a set, but rather in this, in the formulation that it is looking at in that case the optimization problem is a bunch of non-linear equations that need to be solved and that is what it has tried to do ok.

So, with this I think I will stop with my coverage of the Newton method and that is also covers the algorithms that we plan to cover in this course. In the rest of this course what I will now do is dynamic optimization and little bit of dynamic programming to show you how exactly and relate that also to static optimization the kind of optimization that we have studied so far ok.