Convex Optimization

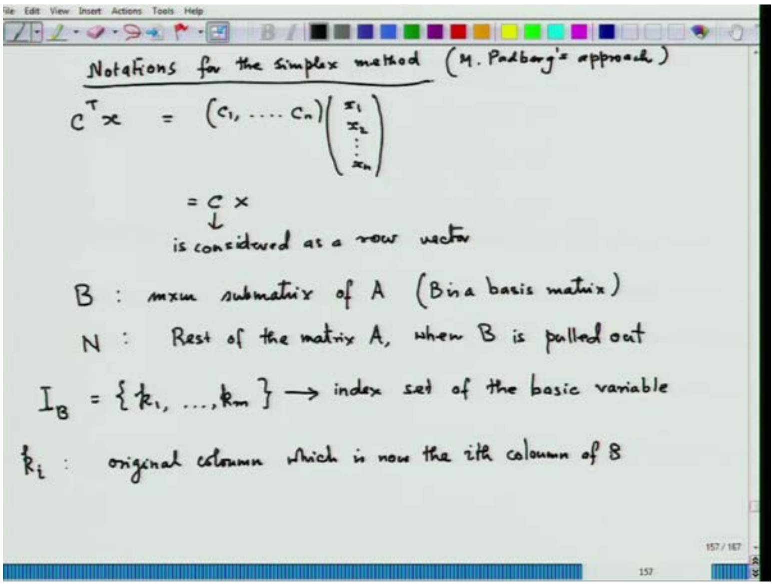
Prof. Joydeep Dutta
Department of Mathematics and Statistics
Indian Institute of Technology, Kanpur

Lecture No. # 25

So, good day to all of you, you can put in place in the day in the sense that you can put in morning, night, evening as you want, depending on when are you seeing this video. So, if you have followed up to the last one then, in the last one we proved the fundamental theorem of linear programming, which says that if my feasible set is non-empty, then there is a basic feasible solution also. And if, I have an optimal solution of the linear programming problem, then there is a basic feasible solution which is, its which is also an optimal solution. So, non emptiness of the feasible set implies existence of bfs, and existence of an optimal solution implies the existence of an optimal bfs.

So, this is a very very beautiful fact, because it says that, the I can always look for what is is. So, simplex method will show you how to go from one vertex to the other in a clever way, so that your function value actually decreases.

(Refer Slide Time: 01:37)



Today, we will keep on a very simple thing to work on. First, we will go, and look into the notations for the simplex method. See the approach that I take again is due to Manfred Padberg, and I take it because I love this approach. Of course, everybody does what he loves. So, here there will be certain simplifications of certain notations, which we will make may be the writing simpler.

We will first write down the notations, and then we will prove two things. First, when I say run the simplex method. How do I know that; I have actually reached an optimal bfs, and I can stop the process. Two how do I know, that I am actually wasting my time. That the problem is unbounded, and I need not bother about solving it. So, these two aspects has to be checked out from certain conditions and involving the problem data. And what are those conditions: number one - a sufficient condition for guaranteeing optimality, number two - a sufficient condition for telling me the that the problem is unbounded below.

So, once I know these two things; it will be easy for me to actually explain you the simplex method. So, we will only study the notations and these two things today, and in the next lecture we will analyze the simplex method. So, when you write C transpose x here, basically what you are writing C 1.

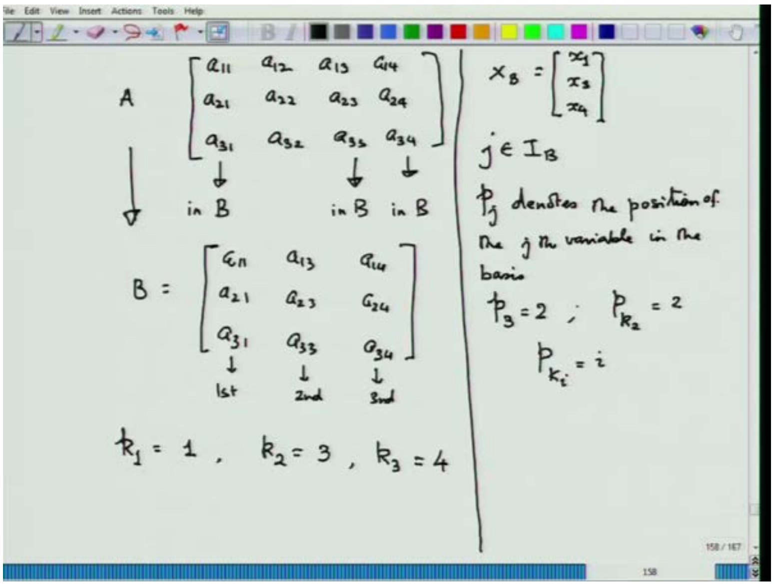
(No audio from 03:02 to 03:10)

I can write this also, as if I take C as a row vector, then I can also write this as C x. If C is considered as a row vector; you know that C transpose is written I write it as C transpose, because this C transpose, because C is usually considered as column vector. For simplicity, we will write C x; even continue writing C transpose x or you can see in a product x does not matter.

So, we will again for simplicity write B to be an m cross m sub matrix of A. Of course, A has rank one, and no non-zero rows or columns. So, m cross m sub matrix of A. So, which is a basis matrix, B is a basis matrix. So, these are notation two N; the rest of the matrix when B is pulled out. So, whatever is whatever though which those columns which are remaining right. So, when you take off the B whatever column is remaining is your N, the non basic part basically.

So, I will denote by I B, do not confuse with it with the I or I x that I have used in the last class. This I B is a you can put some other symbol if you want, but in mathematics we learn not to get confused with symbols. This is the index position - the first index of the basic variable, second index, third index, fourth index. So, this is called the index set of the basic variable; I will explain why - why this sort of strange looking things; why not 1, 2, 3, 4, 5, 6. Now, what does k i c m symbolize. So, k i is a original column which is now the i th column of B the basis matrix.

(Refer Slide Time: 06:19)



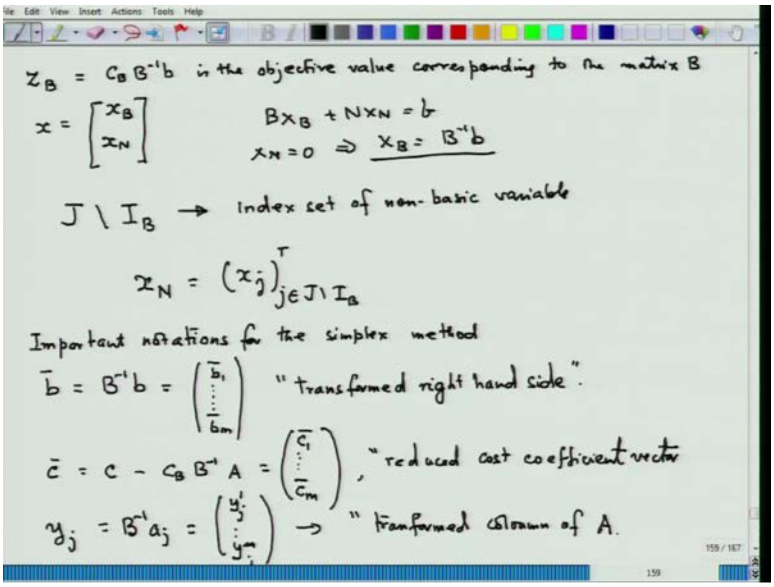
If I just do some explanation by an small example that would be much clear. For example, I have a 3 cross 4 matrix. So, it is a 1 1, a 1 2, a 1 3, a 1 4, a 2 1, a 2 2, a 2 3, a 2 4, a 3 1, a 3 2, a 3 3, a 3 4. Now this is a 3 cross 4 matrix, and if I assume it to be full rank all this rows are linearly independent. So, I can choose any three columns which are linearly independent. Suppose my basis matrix I, this goes in B this goes in B, and this goes in B. So, my B would now look like the following. The stack of the rows nothing else...

(No audio from 07:12 to 07:22)

Now, this is the first position, first column of the basis matrix, this is the second column of the basis matrix, this is the third column of the basis matrix. So my k 1, which is the first position in the first column of the basis matrix was the first column of the original matrix. Now k 2… So, the second column of the basis matrix corresponds to the third column of the original matrix. So, this is the original matrix A, and you get the basis matrix B. So, and k 3 the third third column of the basis matrix is the fourth column of the original matrix. So, this is this is this is what this notation actually means. So, I will just divide this page a bit. Now, x B denotes the vector corresponding to vector whose components corresponding to this columns. So, x B in this case would corresponding to the basis columns.

So x 1 basically, x k 1, x k 2, x k 3. So x 1, x 3, and x 4 this is my basis matrix, and remaining are the non basic part. Now, if I have j element of I B, p j denotes the position of the j th variable in the basis. So, p 3 for example, j th variable x 3 is positioned in the basis is 2, p 3 is two. But if you observe also p, what is 3 here is k 2. So, p k 2 is 2. So, in that sense one can write in general p k i is i. So, tells you that the third variable in the actual setup x 1, x 2, x 3, x 4 is a second variable of the basis matrix. So, these are certain notations which are helpful when we will do the simplex method.

(Refer Slide Time: 10:14)



So, usually Z B is equal to C B B inverse B is the objective value…

(No audio from 10:25 to 10:39)

In the matrix basis matrix B. If you have forgotten why it is like this, you know that I can always divide x whole x into x B and x n see, it should be C B transpose in usual text it will be C B transpose, but we are just considering C B to be a C to be a row vector, So it is written like this. So, I can write this as this; x as x B x n and then you know that A of x is equal to B. So, an A is partitioned into B and n. So, you can write B x B plus n x n is B. And if x n equal to 0, you know that X B and C B transpose inverse B which is same. As whatever whatever we have written, C B transpose is we are not writing C B transpose, because that is taken to be a column matrix, A to be a row vector row vector.

So, Z this is of course, a notation for index set of non basic variables. In this case, what are the example given in the previous page, the non basic variable was x 2. So, if this j this consist of consisted of just two in the last case. So, x N actually can be written as x j in a short form where j belongs to x j transpose in fact, this is. If I write it as a row vector transpose j belongs to j minus I B. So, these are many ways to write the obviously, simple looking things. So, but certain notations are important for important notations for the simplex method.

(No audio from 12:43 to 13:02)

First notation is B bar. This is the transform right hand side or x B basically B y is x B. Because you are operating B inverse on b, so you just write this as; this is nothing but the first part, this is the B bar of the bfs. It also called the transformed right hand side.

(No audio from 13:43 to 13:55)

And now, C bar C minus C B actually books would have C B transpose B inverse A, but as far I want to remind that this is a row vector. We have just taken for simplification simplification of the notation, this is called the reduced cost coefficient - there is a cost coefficient C and I am reducing the value of the cost coefficient by this amount. This automatically comes this thing naturally appears. you might think that how are you suddenly bringing up this fact, bringing up this number, but no actually people try to figure out sudden thing; how to get optimality how to get unboundedness. So, mathematics is always guess and test.

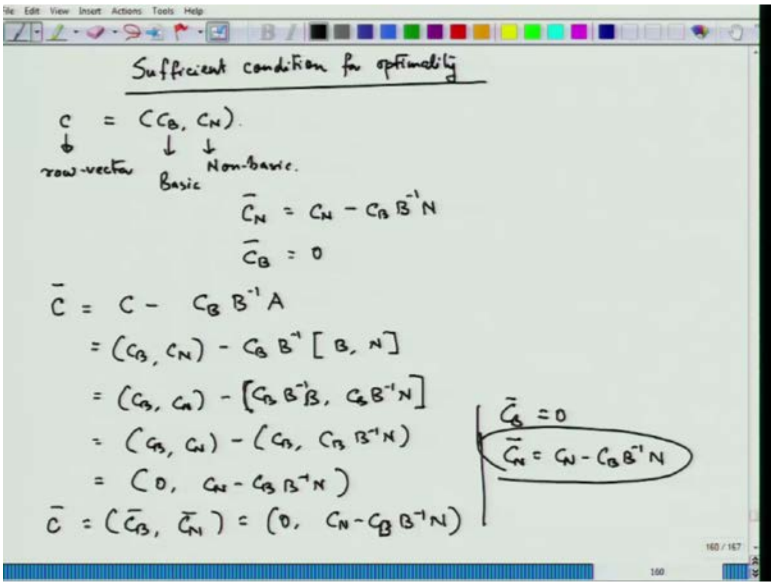
These things are not written very in the very beginning dantzig did came here came to this or rather I would say dantzig really did not bother much about the matrix notation, but when the matrix notation came, what he just wrote it down like this. People were actually trying to figure out under what condition there will be an unboundedness, and the what condition I will have optimality. Then these things appeared quiet naturally in the expression. And hence, conditions had to be imposed on these things. These are these things that we writing down, and that is why people wrote them down separately as important notations.

And most important thing in that everything is just computed from the problem data, there is nothing external that we have to bring into is called the reduced cost coefficient. So, then these names are obviously, given and became (( )); obviously, I cannot pinpoint you who had given this names, but it is usually taken like this.

(No audio from 15:45 to 15:54)

Another notation which is important is y j is B inverse a j. So, their j th column I am multiplying by B inverse, I am transforming the j th column. So, at least y 1 j, y m j where the m elements; so this is called the transformed column of a transformed column j th column. So, it was a j now the transform column is y j.

(Refer Slide Time: 16:42)



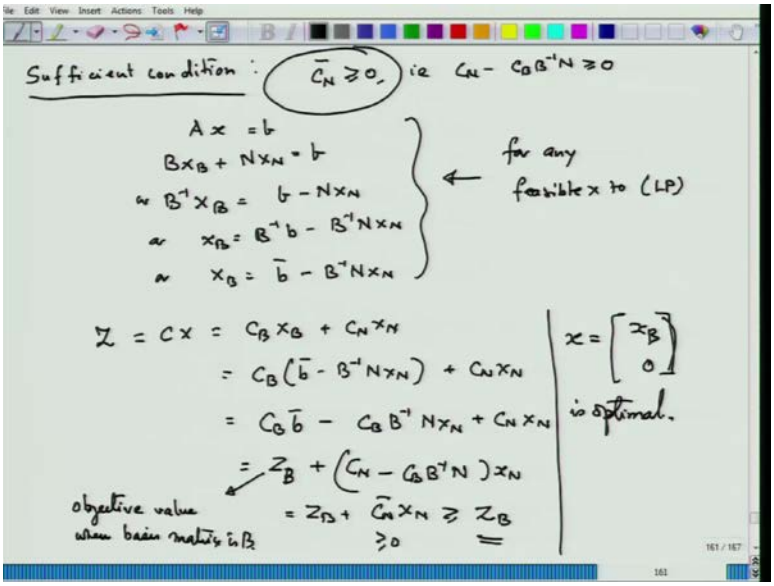
Now, let us figure out a condition for optimality. So now, we will define try to prove that under a certain condition given condition, we will get an optimal solution. We will that is we will guarantee that an optimal bfs has been reached. So, sufficient condition for optimality. Now, this cost row vector I would again remind you, because there people might be so, conversant with writing things such column vector C is equal to C B, C N. So, this part cons corresponding to the basic part, and this is the non basic part.

Now, just look at the cost coefficient in the now reduced coefficient only for the non basic part. Now, it is C B B inverse not if you look at the reduced cost coefficient in the last, it was C B B inverse A, but here I do not want hold away; I want only the N part of A right. So, because I can write this A as B and n. So, I can get the whole thing divided into two parts. So, one will be C minus this, then and another will be C minus that. So, I I I just this C would C B minus that C n minus this, because this can be written as C B minus C B, C N, and this can be written as C B B inverse capital B. So, which is I. So, it will become C B minus C B and and then I would have C bar N. So, I will just work it out carefully.

So, now you might ask me what about C bar B, because that would give you C bar. The total reduce cost coefficient it is 0 actually. How, because you see I can write my total reduce cost coefficient as C minus C B B inverse A. So, this can be written as just as above C B, C N minus C B B inverse A is divided into B and N. So, this can be again done through simple matrix manipulations, C B B inverse, so this is be a vector which consists of C B B inverse B C B B inverse N. So, this is C B B inverse be the identity matrix. So, C B into identity matrix is identity. So, it is C B, C N minus C B B inverse N.

So, zero C B minus C B and C N minus C B B inverse N. Now this is C bar. So, C bar if I write as C B bar, and C N bar which is equal to 0 C N minus C B B inverse N. So, we will immediately have C N bar, we will immediately know that C B bar is 0, and C N bar is C N minus C B B inverse N. So, it is a reduced cost efficient associated with the non basic part which gains more importance in telling you whether the bfs - current bfs is optimal or not.

(Refer Slide Time: 20:55)



So, let us see. So, what is that sufficient condition. Sufficient condition is the following one - sufficient condition is C N bar is greater than equal to 0, that is C N minus C B B inverse N is greater than equal to 0. So, this is my (( )), if I look at the reduced cost coefficient corresponding to the non basic components of the bfs. Then if all of them are greater than equal to 0, then I have reached my optimal solution; that is the current bfs is optimal. Now, again go back to writing A x equal to B which can be written as using the partition B x B plus N x N is equal to B or x B is equal to B minus N x N or x (( )) x B inverse x B.

So, x B is equal to B inverse B minus B inverse N x N. Now, observe that B inverse B are we have already given foundation B bar or x B equal to… Now, Z consider any feasible x. So, this is true for any feasible x.

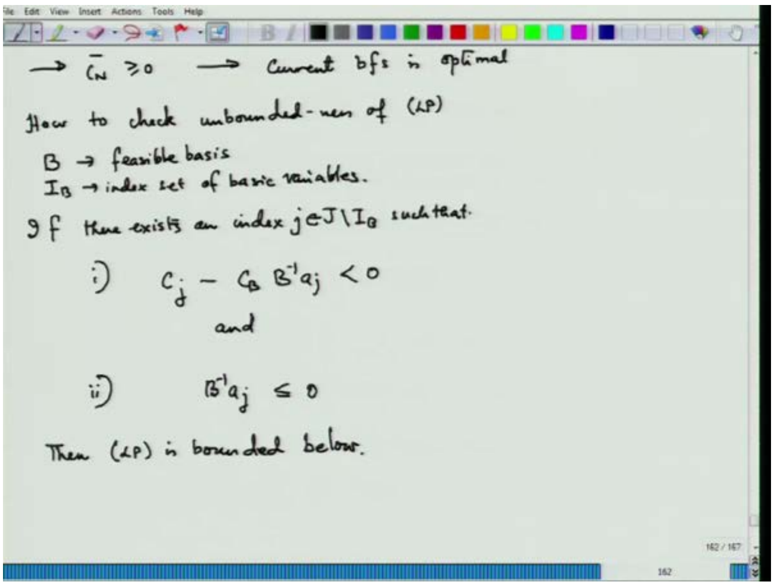
(No audio from 22:38 to 22:47)

Where x and C. So, let me compute Z of C x - this gives you C B x B, C N x N just the separation. C B what is x B; it is B bar minus B inverse N x N plus C N x N. So, I will write C B B bar which is B inverse B minus C B B inverse N x N plus C N x N; for C B B inverse B is the optimal solution, which I call Z naught. Z naught is the optimal solution at sorry yeah, z naught is the why I should write z B. So, z B is the optimal value to the current basis B. So, z B is the objective value or not optimal value I making mistake; (( )) objective value when basis matrix is B which is the case.

Now, once I know this this is I can now this combine these two, two right plus C N minus C B B inverse N x N. So, what I get is z B plus C N bar x N - x N is a greater than or equal to zero, because x x is feasible and C n bar is greater than equal to 0. So, whatever feasible set you take, the for feasible whatever feasible element you take from C, what that is whatever x you take from C the feasible set to LP. The objective value at that feasible point C x is always bigger than equal to this value Z B. So, and Z B is equal to C B into B inverse b. So, Z B is actually an optimal solution, and hence the current basis B is the optimal basis and x equal to x B 0 is the optimal solution.

Because you see if I put x N equal to 0, then Z is equal to Z B. So, and and that corresponds to the x when x where x N is equal to zero right. So, here you see I proved that if this, because C N is greater than equal to 0. Because of this condition Z B is a optimal value, and Z B is a is a optimal value why, because Z B is the value of Z at x B which is B inverse B, and x is any feasible point. So, the optimal value at any feasible point is bigger than the optimal value at the particular bfs, and hence the particular bfs is optimal. So, it implies that x equal to x B 0 is optimal.

(Refer Slide Time: 26:16)



So, you know how to check optimality - optimality is check by checking. So, if I just check C N bar is greater than equal to 0, I know current bfs is optimal. Now, how to now next question is how to check unboundedness. As a problem itself does not have a solution, how will you check the and the beauty is that, there are methods to check it directly from the problem data; that is why linear programming is so beautiful. So, as before B is my feasible basis I B is the index set of basic variables. Now, we will be really concerned with the non basic part.

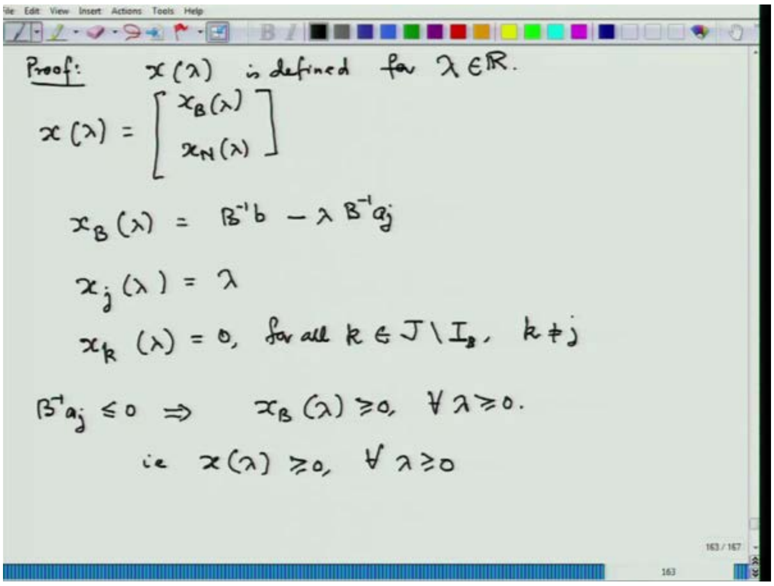
If there exists an index in j element of j set minus I B, that is the non basic part. Such that C j minus C B B inverse a j. So, basically this is the reduced cost coefficient corresponding to the j th position, the j th vector in this set. This is strictly less than 0; showing that optimality is not reached. That is C C bar N cannot be greater than equal to zero, if one of its these vector C bar N greater than equal to zero means, all its components are greater than or equal to zero. But if one of its components are strictly less than zero it cannot be. So, current bfs cannot be optimal; C N greater than equal to zero implies current bfs is optimal right.

If current bfs is not optimal, you will also get this right. So, if current bfs is not optimal, this will always occur. This is very very important that there would be some j for which this would occur, if current bfs is not optimal. Because this implies current bfs is optimal, if current bfs is not optimal something like that we will occur. So, if current bfs is not optimal this will occur. So, one this sort of thing is occur, we should be see that here there is a creek right. This is not a if and only if condition; that if the current bfs is optimal, then this will always be true.

It says that if this is true the current bfs is optimal right. So, it is not a necessary condition, but if I have one of these to be that the this is greater than not greater than zero. Then we cannot be sure whether the current bfs is not optimal, but if the current bfs is not optimal it will give you something like this. So, once we have something like this in the algorithm, we try to be cautious; we know that possibly we are not is the optimal and make a try to go to the next next point. And suppose these conditions are satisfied one and two; the transform j th column that is y j which is B inverse a j is less than equal to 0.

So, there is a particular index j for which this happens; if this two happens then LP is bounded below. This is a very very important result, such by checking the problem data you can tell just at by looking at the problem data you can tell, whether a problem is unbounded just after (( )) upon iteration. So, how do you go to prove this. So, our next effort would be to prove this.

(Refer Slide Time: 30:43)



Now how would we do the proof. What we will do is that we will find a sequence of points or continuous gathering of points, we find a sequence of points, such that those sequence of points is feasible. At the same time if I compute the objective values at those particular points then my objective value will continuously decrease, and continuous keep on decreasing without n, and go towards minus infinity showing that my problem is unbounded.

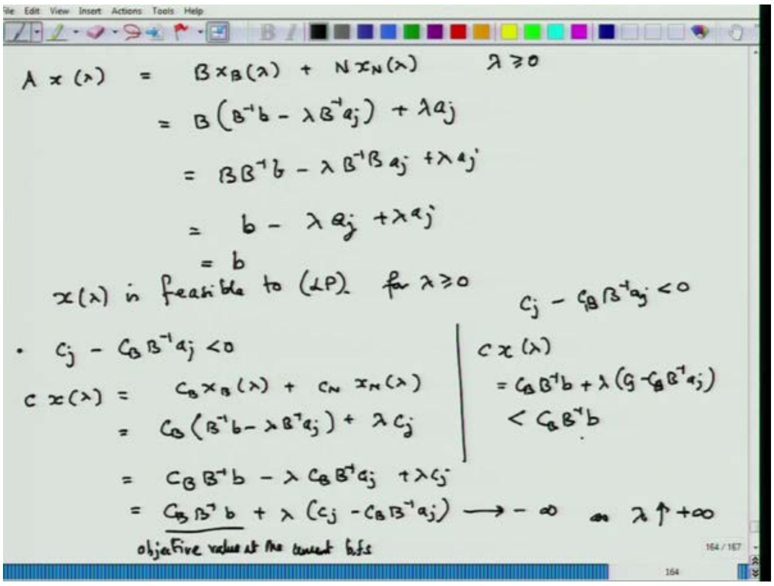
So, I define of the x lambda is defined for lambda in real number line R. So now, x lambda is also partitioned into two parts. In the same way, x is partitioned. I partitioned as x B lambda and x j lambda sorry x N lambda. These are way you should partition it. So, x B lambda I define as follows. Now, x N lambda has two parts or J minus I whatever n instead of n you could write J minus I B whatever does not matter; I I am just n is a better symbol. So, what happens when when I am in the non basic part. So, then x j is lambda - x j lambda is equal to lambda and x k lambda is equal to 0 for all k which is in j, which is in this index j minus I B, which is in the non basic part right n basically this is, this is n if you want to write. So, and k is not equal to j when k is equal to j that is then x j lambda is lambda, otherwise this is zero.

Now, B inverse a j, I assume to be less than equal to 0. So, this is negative and there is a negative sign. So, this makes it positive. Now, B inverse B x been a feasible element right. So, B inverse B is a value of the objective value at the current basis. So, if you take, if you have taken any x which is optimal; obviously, then B you can in B inverse B is the x B part. So, there is a bfs and there is a x B part. So, if B is a feasible basis. So, B inverse feasible basis means B inverse B is greater than equal to 0 by very definition. So, which is basically nothing but the x B of the bfs associated with this basis B.

So, this this is greater than zero, this is negative negative, this is less than equal to zero and there is a negative sign. So, this makes it greater than equal to zero and this is greater than equal to zero, because B is a feasible basis. So, this would imply that x B lambda is greater than equal to 0, for all lambda greater than equal to 0. So, that is in general x lambda, because this is lambda greater than equal to 0, and this is zero is greater than equal to 0 - for all lambda greater than equal to 0. So, when lambda is greater than equal to 0, this negative and negative of this makes it non-negative, and this is already greater than equal to zero, where B is a feasible basis, this is what we have.

Now, we have to prove. So, we have proved one part of the feasibility of x lambda. So, we will show that if we compute the objective values, at at each of these points, the objective values will continuously decrease as the lambda goes towards infinity.

(Refer Slide Time: 34:51)



So now, I have to compute A x lambda. To show that x lambda is greater than equal to zero is guaranteed; now, I want to show that it is also feasible. So, B x B lambda plus N x N lambda. So, this is nothing but B what is x B lambda? x B lambda by definition is B inverse B minus lambda B inverse a j. So now, for me now lambda is greater than equal to zero, because that that is when it is feasible. So, our lambda now is obviously, greater than equal to zero plus N, and now x n has two parts. So, at the j th part it is zero and at the j at when at the j th part j th component is lambda (( )) is zero.

So, basically N into x lambda will give me lambda into a j, the j th column in the nth part. And the remaining part is obviously, zero. So, here I have B B inverse B minus lambda B B inverse B a j plus lambda a j. So, this will give me B B inverse is B minus lambda A inverse B is identity a j is a j plus lambda a j which is B. So, x lambda is feasible; now, obviously, for lambda greater than equal to 0 for any lambda greater than equal to zero. Now, we are given the following that C j minus C B B inverse a j that is the reduced cost coefficient one of our j th column, corresponding to the j th column is strictly less than 0.

Now, what is C or x lambda I want to compute this. That is C B of x B plus C N of x N. Now, this would give me C B x B lambda is again B inverse B minus lambda B inverse a j plus C N x N, x N what is your x N? It is only lambda at the j th position and all other positions it is zero. So, it will become nothing but lambda times C j corresponding j th position here. So, C B B inverse B minus lambda C B B inverse a j plus lambda C j. So, this is C B B inverse B plus lambda into C j minus C B B inverse a j. Now, I know that this is strictly less than zero.

Now this is fixed, this is the objective value of the current basis value at the current bfs. Now this is negative, now if I make this positive I can keep on making positive x lambda would still remain feasible, but this value would go to minus infinity. Because this is negative, and this is positive; negative fixed negative number, and this is the positive number and making large and large. So, this will become negative and negative and continue to go down, as lambda tends to plus infinity goes up to plus infinity.

So, with this we end today's talk here, we have two interesting conditions. One tells you when you know that you have optimality, and one another tells you that if this, and this both are true then you have a… So, this is true that if one of this is strictly less than zero, you know that current bfs is not optimal, because you take one lambda and I show that this is negative. So, this whole thing is less than the C. If C j, C B B inverse a j is strictly less than 0, if it is given to me then C of x lambda for the particular lambda is equal to C

B B inverse B plus lambda times C j minus C B B inverse a j, because this is negative and this is non-negative.

Suppose I take lambda strictly bigger than zero, then also x lambda is feasible. So, for such a lambda what I will have here is that this is negative and this is positive. So, this will whole thing would be negative. So, this whole thing would be strictly less than C B B inverse b. So, I have found the feasible point whose object value is strictly better than the objective value at the current bfs. So, the current bfs cannot be optimal. So, when you are doing the simplest method. Once you find the j column of the negative of the non basic part at which the value, the reduced cost coefficient is negative; you are sure that your optimality is not reached. You can have a feasible point whose objective value would be better than your current bfs. So, you have to go for a new one; then you would check whether B inverse a j is less than equal to zero.

For all the columns in which you have this, all the columns in the non the non basic part when you have this, you have to check also whether B inverse a j is less than equal to zero; then if there is one j for which this is true, then that will be the guarantee that from this result that the problem is unbounded. If none of them, if all of them are strictly bigger than zero, then we do not have an optimal solution, but the problem is not unbounded and we can proceed; problem need not be unbounded and we can proceed. Not we cannot guarantee its unbounded not unbounded from this the there is no information which tells me that it is unbounded. So, I can then try to proceed.

So tomorrow, we will start with the exact procedures or the updating techniques used in simplex method. What is what does the simplex method do; we have a current basis if it is not optimal, go to a new basis where where which which could be a optimal or where the function value is we have to going such a way, that the function value is strictly less than the function value earlier. So, that is exactly what we would show you in our next lecture, thank you very much.