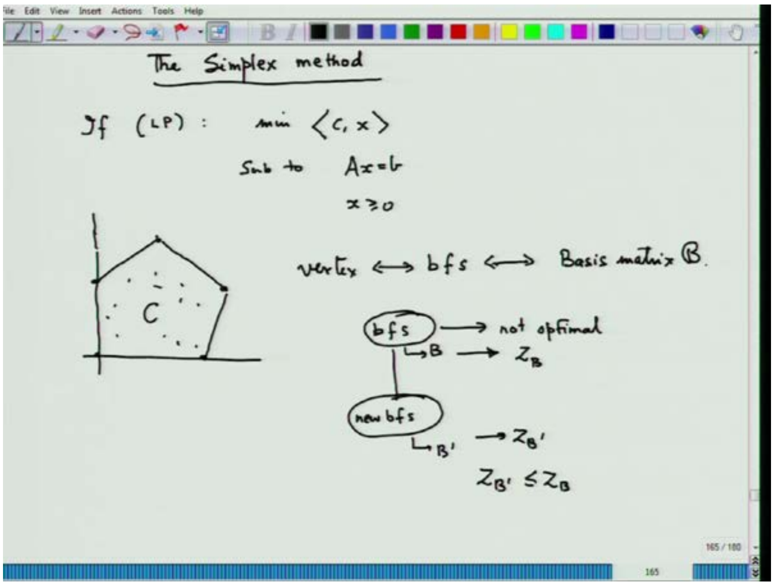
Convex Optimization
Prof. Joydeep Dutta
Department of Mathematics and Statistics
Indian Institute of Technology, Kanpur

Lecture No. # 26

Today, we will start discussing simplex method; in the last one, you know that we have shown you conditions under which, you can check whether the problem is unbounded, you can check whether there is a condition, whose satisfaction would give you the optimality that you can guaranteed that the current bfs is optimal. So, we start basically with the philosophy of the simplex method, which is what we have going to build upon. Any mathematical method would have a basic philosophy; philosophy not in the sense of what philosophers are telling, but philosophy in the sense that what is the basic frame work that we have to work upon, what we intend to do, and how do we intend to do it; and that is exactly what we are going to concentrate on.

(Refer Slide Time: 01:13)



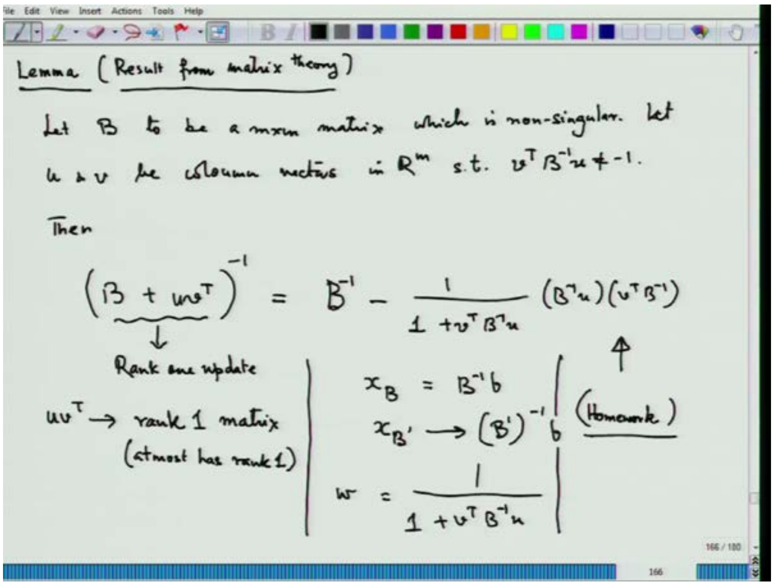
So, what we have understood that if, so if LP that is this problem, minimize C of x subject to A x equal to b and x greater than equal to 0; if this problem have a solution, then there is a vertex on the feasible polyhedron, which is also a solution. So, which means that if I have the feasible region, say like this; this is my feasible region C, it is a convex polyhedron; then what I can do is compute the objective value that each extreme point, and check which is the minimum, but if the number of extreme points is very large, which is often the case, then it is very difficult to make such computations, and make a direct enumeration.

So, what would we do is that we would rather try to find a clever way that if I have a current, I am I am on a vertex, and if I know that this vertex is not optimal, then I have to find a clever way to go from one vertex to other vertex so that my objective value decreases, because I want to minimize the function. Now, how do we do that? So, we have to understand that every vertex corresponds to a b f s basic feasible solution, and every basic feasible solution corresponds to a basis matrix B, which is a m cross m matrix of rank m corresponds to some basis matrix B.

Now, this is very, very fundamental. So, if my current b f s is not optimal, then I have to move to another b f s, move to a new b f s. So, if I call this, basis matrix associated with these b f s as B, and this b f s as B dash, and the objective function value, when the I take this b f s as Z B, and this b f s as Z B dash, I should have at least… This is what I intend to have; this is this is basically some sort of flow chart of a linear programming problem of the simplex method; and that is what I want intend to do in the simplex method.

So, how do we achieve that? So, our goal in this talk today, which would we basically the end of this sort of simplex type approach to linear programming, because we are this is a only a short patch up inside this larger course of convex optimization. Now, what we are going to show now is how to do this change from B to B dash. So, the whole story of simplex method lies in understanding how to make this change; and what lies behind making this change.

(Refer Slide Time: 04:48)



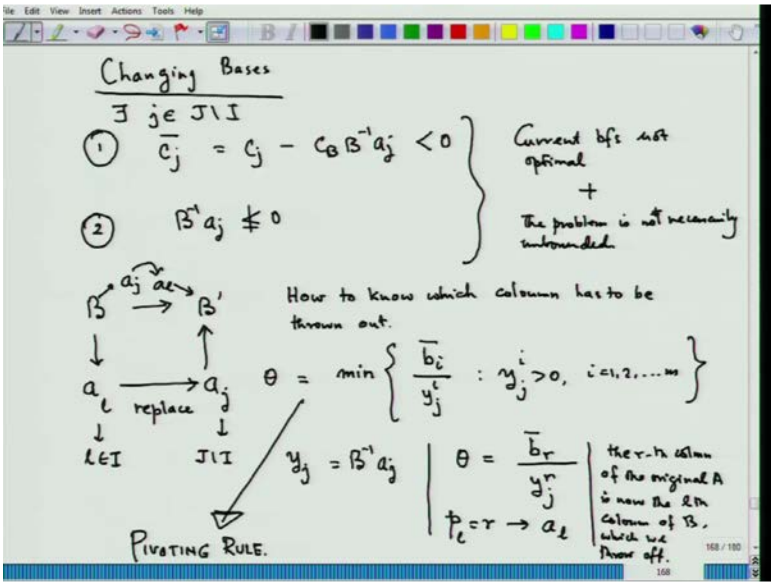
So, as we start to do this, we first start with the lemma, which is essential to our proves of the main result, so this lemma is a lemma from matrix theory. Now, I will not prove this lemma, but ask you to do in the homework. So, what is this lemma? So, let us consider B to be a m cross m matrix, which is non-singular that is of rank m means it is invertible; once I have this, let u and v be column vectors, we are just making this distinction, because we are using c as a row vector in this description. So, this the column vector in R m, such that v transpose B inverse u is not equal to minus one, you can always choose such vectors, you can always make such an choices.

So, if this is the case, then… So you should observe that v transpose B inverse u, so this is what you have. Now, you have to understand why we are computing this inverse; this is this whole thing this B plus this part, B plus this one, B plus u v transpose is called rank one update; this is called rank one update, because u v transpose, this matrix is a rank one matrix; I cannot say it is a rank 1 matrix; what I can say at most as rank 1, it can rank cannot be more than 1, but does not matter if you just take it to be rank 1 matrix, so without loss of generality, such things can be said.

So now, why I need this inversion business? The inversion business is very important, because how do you find x B, see even it start, there is B inverse B, so anyhow the new one to find x B dash. So, find a x B you need B inverse, which you will all anywhere know. So, to find x B dash, which is corresponding to the new basis matrix B dash, you

you need to know B dash inverse, so I have to find the formula, which automatically computes the inverse. So, if if it automatically computes the inverse, it is easier for me then rather than having this matrix v plus u v transpose, and then trying to inverse take the inverse by the usual matrix inversion formula, but rather have a simpler formula which will do this; and this thing will always get satisfied when we have linear programming problem, because you see expression is well well defined, if I call this expression as w, if this this is minus 1 not equal to minus 1, so this cannot be 0. Now, this as I have just told is well defined, so this formula is well defined. So, how to prove this formula? This you do in the homework; may be some tips can be given, but later on, not now. So, our next step is to know how to change bases and this is exactly what simplex method does.

(Refer Slide Time: 09:22)



(No audio from 09:20 to 09:29)

What do you do when you change a bases? You have a basic matrix B, when you take you replace one column of the matrix with some other column, you bring one column from one basic part, and put it in the basic matrix, and take one column from the basic matrix, put it in the non-basic part. So, then you have a new matrix, which is a basic matrix. Now, we have to understand, when will you start changing the basis? What are the two conditions which will lead you to start changing the basis? So, number 1: You have to first find that there exists a j, j element of the non-basic part, j is 1 to n, and i is the basic part, and the other part is on basic part. So, there must be a j for which this should happen that is I am not sure whether my current b f s is optimal.

Number 2: You have to be sure that the problem is not unbounded; this need not be less than equal to 0. So, two things has to be guaranteed current b f s not optimal that is the b f s corresponding these basis matrix B is not optimal; last, we have to guaranty, which is the second one, the problem is not unbounded. See, this is not telling you that okay, I am guarantying that the problem is not unbounded, it is telling you that okay. You have not been able to get the sufficient condition, which tells you that the problem is unbounded. So, we are assuming that… For the time been now we have no condition which is telling me that the problem is not unbounded. So, we know that the problem could be unbounded also, but the problem at the moment we know that we have no way to check whether problem is unbounded. So, we go and proceed to the next step.

So, the problem is not necessarily unbounded. So, this is these are the two points you require. So, we go from B to B dash; how do we do? So, we take a column a l of b, choose a column a l of this b, and replace this column by a column a j, where j is from j minus i, and l is from i, so, this l is of course, from i, and this j is from j minus i. Now, once you put instead of you swap a j with this, you are swapping a l with a j, then then the new matrix that is formed is basis that is the basis matrix, means you swap a j with a l, and then that is exactly what you will get.

Now, how to know which column has to be thrown out?

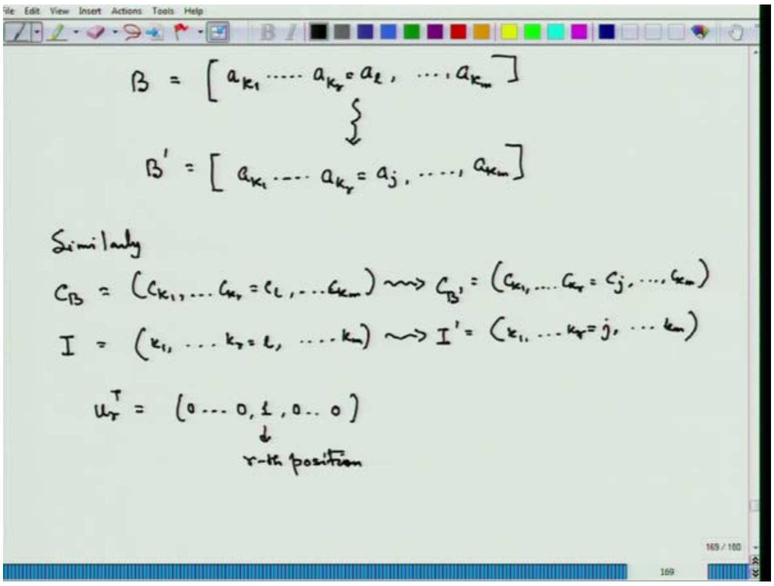
(No audio from 13:06 to 13:16)

This is important; how do you know that which column you need to throw out? So, to know this, you do a little trick, the trick is as follows. Now I will not go back and recollect with you, what were the symbols; for example, I will use the symbol B bar; now and you possibly would know that the symbol B bar is already known to you, we have already given some symbols early in the course, and B bar is B B inverse B. So, we are not going to go and repeat these symbols anymore.

So, what you do is, you compute this theta, find the minimum of; so, now what is y j? y j will use a notion called y j, which you already know; so, i eth component of the j eth column, so y j is B inverse a j, so i eth component of the j eth column; now I know that this is not less than equal to 0. So, there is some i some some column, some component of this vector B inverse a j that vector must be strictly bigger than 0. So, there is at least 1 y i j, which is strictly bigger than 0, y i j is the i eth component of the j eth column, where i is running from 1 to m. So, i goes from 1 to m; so you just consider those i (s) is for which this is bigger than 0, and then divide by this, and take the minimum; and let, so because there only finite number of such thing you have, one of them to be the minimum. So, theta let that be equal to… So, if p l is equal to r, then it is the a l, then a l, so let p l be equal to r; so it is the l eth column, so r is the corresponding column of the actual matrix, where l is the corresponding position of that particular column in the basis matrix.

So, you know if I have p l is equal to r right, then it is the r eth column of the original matrix, which is inside the basis matrix now, which is taken he l eth position in the basis matrix that has to be replaced off. So, the r eth column of the original a is now the l eth column of b, which we throw out; once you know this, how to do it? Of course, we will tell you, why this is making sense right; now this is the rule, this is called the famous pivoting rule, this rule is called the pivoting rule, which is central to simplest method this is nothing but gaussian elimination this is solving linear equations, but why this make sense; why this rule actually does; what it does and achieves the aim that you want to have that you have a B dash such that z B dash would be less than z B; in order to do that, we need to go bit further. Let us make some diagrammatic re presentations.

(Refer Slide Time: 17:30)

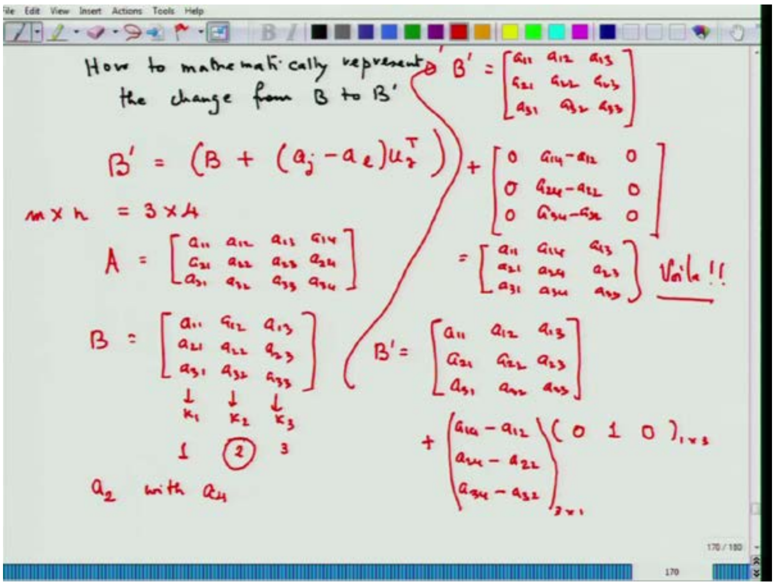


So, what we had been just speaking about and. So, let B I represent the basis matrix. So, its column a k 1 a k r is equal to a l. So, here just I made a mistake I just have to change this is a l eth column of the original matrixes r eth column of the basis matrix, which we throw off. So, we representing the r eth column of the basis matrix; now r eth column basically k 1, k 2, k r is is the l eth column of the original matrix; now you make the change, so new one; so it is a k 1, and in the r eth position is now is becomes the j eth column of the original matrix takes of the r eth position of the basis matrix.

So, I am changing one particular position of the basis matrix with some other column from the original matrix; similarly, if you have done this change, you can done the corresponding changes, you have C B, this is an n number of columns, we are bother about, because is m cross m matrix. So, this is changed to C B dash, which is exactly this; right so, the index set I of the basis matrix will now change into this form, it was like this, this change to I dash which is everything, but changing the r eth number with l here is j, so the r eth position has been changed; if you look at it, you can represent I dash is what; I dash is nothing but you had the I, from which you have taken off l basically, and you have added up j is one way of looking at it.

Now, technically it is not very fine, because I am representing I with k 1, k 2, k m just not bother about it. So, let me define this matrix sorry a vector, a column vector, but I am writing it as a row, so I am putting the transpose, which is 0 everywhere except the r eth position, which has the value component value one. So, this one will help us a lot when we do the change. Now, I define, I want to define B dash, let me see how can I putting this change that I have done form B to B dash, how can I mathematically represent?

(Refer Slide Time: 21:15)

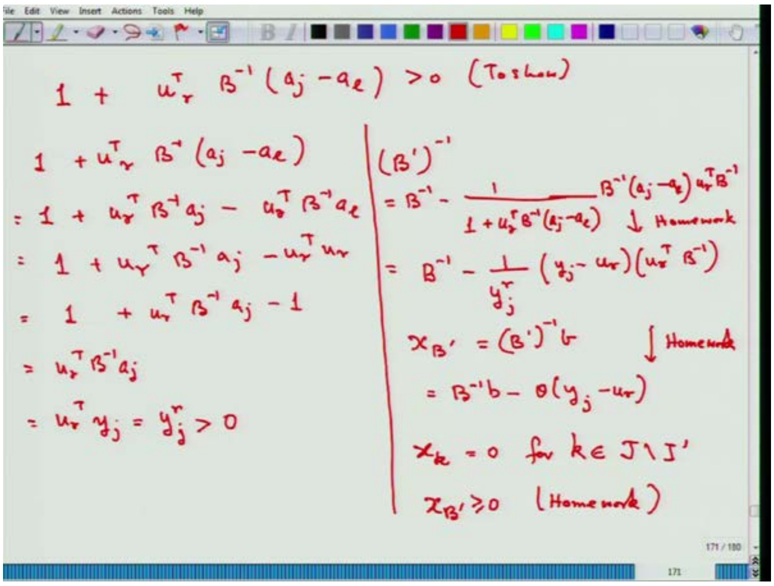


So, how to mathematically represent the change from B to B dash? To emphasize this change I am taking the red color of the pen. Now, I am writing B dash is B plus a j minus a l, so this is a matrix also. Now, if I look at it let me take a simple case for c variable, it is a say a 4 by 3 matrix; say if I take m cross n sorry not 4 by 3, a 3 cross 4 matrix. So, then what does what how does B dash will look like? So, B is nothing but say a 11 a 12 a 13 a 14… a 41 a 42 sorry sorry sorry sorry sorry sorry 3 it is a 3 cross 4 matrix, no sorry this is A, I am writing A. So, A is a m cross n matrix, this is my A; suppose my B has to be a 3 cross 3 matrix; suppose my B is this, a first 3, is my B right which is m cross n matrix. Now, this is my k 1, this is my k 2 and this is my k 3. So, k 1 is 1, k 2 is 2, k 3 is 3. Suppose I am replacing k 2 that is a 2 with a 4. So, my B dash would be B which is a 11 a 12 a 13 a 21 a 22 a 23 a 31 a 32 a 33 plus a 4 minus a 2 right; what is a 4 minus a 2, a 4 minus a 2 is a 14 minus a 12 a 24 minus a 22 a 34 minus a 32 right this is what is, this is the column vector, this is multiplied with this row vector whose r th position, here the r th position is k 2, r k 2 is the r, so it will have 0 1 and 0. Now, I am doing the multiplication which I will carry out here, because I need everything to be in one page for you to… See if I carry out this matrix multiplication what will I have. So, it will be B dash is B which is which is a 11 a 12 a 13 a 21 a 22 a 23 a 31 a 32 a 33. So, you see this is the rank one updating formula if you remember this is the rank one update.

So now, if I do this matrix multiplication, I will have what? I will have the matrix this is our rank, this is our 3 cross 1 and this is 1 cross 3; see we will have 3 cross 3 matrix which is what we required. So, what would be that 3 cross 3 matrix. So, you take this one multiply with this, multiply with this, multiply with this. So, it will be a… So, it will be 0 a 14 minus a 12 and 0, then you multiply with this, this, this will be 0 a 24 minus a 22 0 and then you will be multiplying this with this, will multiplying with this all of this, so a 34 minus a 32 0.

Now, if I add these two matrices - these two matrices see from here I gone up here right. So, now if I add these two matrixes what you will get? When you come here you see this, this, this all cancel with this, this, this and so, this will give me a 11 a 21 a 31 a 14 a 24 a 34 a 13 a 23 a 33 and then as the (( )), this is exactly what is required. This is exactly what I wanted, change the second column by the forth column and that is done by using this rank one update. But if this is my new basis matrix I have to know how to find its inverse which you will know from the rank one update formula, which we have just presented, and also now you have to show that the set B that you get here, set B dash is actually having a lower function value than what you have with set B. Now, let us compute the B dash inverse which is required to compute set B and all those things x x B dash. Now, I will keep on writing with the red pen; to emphasize how important this part is. So now, we have to compute the rank one update.

(Refer Slide Time: 28:47)



So, to compute the rank 1 update, I have to first show this quantity; this quantity is not equal to minus 1 or this is strictly bigger than 0. So, let me this is to show. So, how do I compute this? So, I just again write; so, it is 1 plus u r transpose B inverse a j minus u r transpose b inverse a l; now what is B inverse a l, what is B inverse a l, that is something you need to understand; a l is the thing that your replacing a l is actually, if you look at it a l is the column that your replacing.

Now, if you look at the basis matrix. a l is the one of the column is the r eth position in this basis matrix; now if if you do B B inverse b, you get identity the identity matrix B inverse b, which you know very well. Now, how what would happen, if I operate the inverse - b inverse on the r eth column of b, it will give me the r eth column of the identity matrix, which is nothing but u r in this case. So, this is equal to 1 plus u r transpose B inverse a j, but this just 1; so I will cancel these two to get. So, what is B inverse a j? B inverse a j is y j by definition; we have already mentioned many, many times that B inverse a j is symbolized as y j; now what is this? This is y r j; the r eth component of the j eth column of inverse a j, this vector, r eth component.

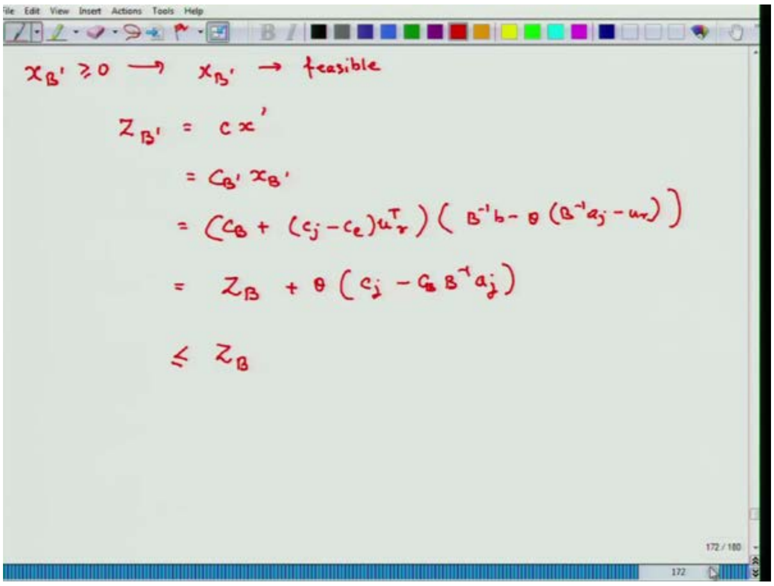
But by the pivoting formula, which we have got by the pivoting that we have done here, this pivoting tells me that this time thing is strictly bigger than 0, and hence we have proved for we wanted; now we use the updating formula to compute the B inverse sorry B dash or B dash inverse, which by the updating formula is B inverse minus…

(No audio from 32:09 to 32:38)

This is exactly the formula right. Now, this with this formula I can conclude, which you might be asking certainly what is happening, but sometimes you need to works things out to have the actual fund of the subject. So, this to this is your homework, because y or (( )) is strictly bigger than 0, then this is the well defined as we have seen, so this is the well-defined thing, so this exactly the formula.

Now, I can now write x B dash x B dash is B dash inverse b. So, B dash inverse b you can again compute out to be this; and x k is equal to 0 for k, which is in j, but not in I dash; now how do you come is again homework, which will be part of your homework assignments that will be given at the end of the course; this will be though I am mentioning them as homework, homework, homework, this will also come in a nice form at the end of the course. So, you can try them out and you will have my email address at the end if you are having a problem, you can just let me know and I can send you the details.

(Refer Slide Time: 34:33)



Now, once you know that is x B dash, you can prove that x B dash is also bigger than 0; if you go here, from here you have to prove that x B dash is also bigger than 0; now how do you prove that x B dash is bigger than 0 would be your concern; of course, B inverse b is bigger than equal to 0, you have to prove that this is actually negative. So, this theta is positive, so you have to prove that this is negative to prove that this is bigger than 0. So, your job is to show that this is bigger than 0, it is also homework. So, I have given you quite a bit of homework this is a very, very important thing that is why you need to compute out this, because while you compute out this, you have a lot of understanding of the mechanism of simplex method and that that every student needs to do it himself.

Now, you compute, so once you know this, you know that x B dash is feasible, and it is a bfs by your definition; now you write down Z B dash is C x dash, but by definition, by our notations, it is this. Now, what is C B C B dash? It is nothing but the same formula for the B, c j minus c l, this into B inverse b minus theta B inverse a j minus u r; now you will compute to find the following, so I can you will be able to write the whole thing as with this into this is Z B. So, you can write the whole thing finally as… if you look at this whole thing, here this j eth row the j eth row, which is entering the scenario, entering the basis that has a pivotal role to play, because we are computing B inverse y j, B inverse a j is equal to y j using that particular row a j, that sorry that particular column a j of the matrix A.

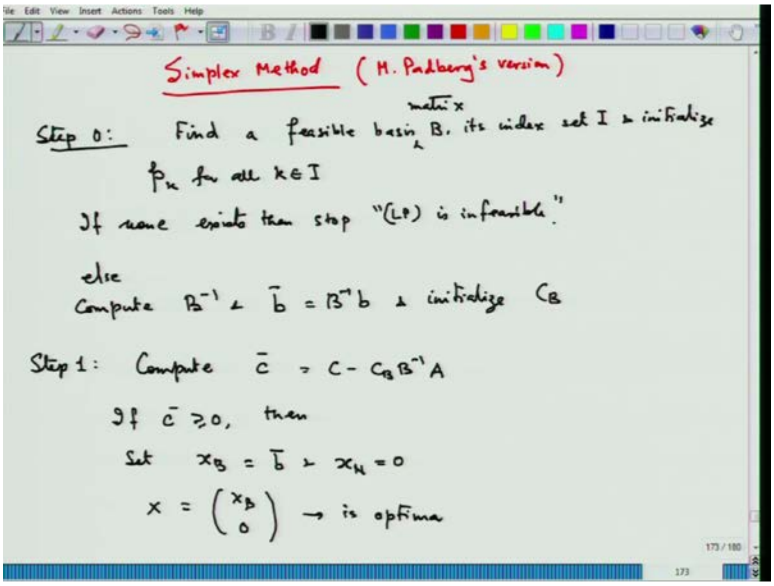
And then we are computing the y r j. So, this is very, very important the theta is very important, because to this theta is computed only when you know j, and j is known, because you have a particular j, for which you have those two conditions that the sufficiency condition for optimality is not valid, sufficiency condition for unboundedness is not valid, if you have that particular j eth column, then you use that to do the pivoting; and you know that which, then that is the column that will enter the basis, that you are sure, but knowing which will enter the basis matrix, you use that to find that which will get out of the basis matrix, and you this fact is actually working, because it is giving you the new matrix and now we will show that we can show that the objective value decreases.

So, this thing you know has been strictly less than 0 in the beginning. So, this is the j that we have been working with; and see this theta is positive, so this whole thing is to get this strictly less than 0, and theta is greater than equal to 0, this is less than equal to z, and that is what we wanted. See observe that, our first starting has been with this. So, I am giving you a j, where this is satisfied; please do condition, and with that j you start working with that particular j, because you know this you know that there is something like this, and then you compute theta; once you compute the theta, you get one particular component r and corresponding to that r that r has to be one among the m components.

So, that corresponding component is the particular position in the basis matrix, particular position column position in the basis matrix which has to be thrown out and this a j has to be put in that is that if it is r is equal to p l, then you are throwing of the a l eth column of the matrix, which is in the r eth position of the of the basis matrix. So, it is the j that starting j that we have we are working with that j throughout, so that is, so, we have told you how to make the update from B to B dash; how what is this inversion formula; and actually we are basically computed x B dash. So, this theta is the pivotal rule, because at the end, I am not really bothered with B or B dash, at the end I am only bothered with what? I am only bothered with x B and x B dash; if I know x B given to me in the beginning starting basis matrix that is why we always take the starting basis matrix to be identity that is what we do in the standard implementation of the algorithm, which we will not talk about here; there many, many there is a separate course here which talks about how to implement the algorithm. And we have shown that how can we earlier that we can introduce slack variables and get basis matrix.

So, if I have a starting basis matrix B, if I know that basis matrix, at every step I can just compute the theta. So, this theta once I have computed, I will immediately know x B dash. So, but it is very important to know B dash inverse also, because this B dash inverse would be used to go say for B double dash inverse or B x B double dash. So, if you go from B dash to b double dash this b dash would become very, very important and as a result of which these two steps are very important. So, we have basically done the simplex method; now and we show that our approach actually solves this.

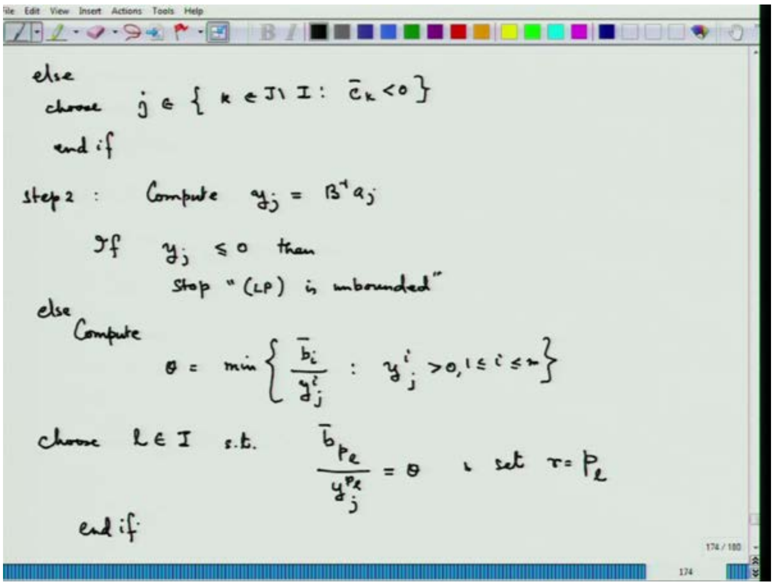
(Refer Slide Time: 41:13)



Now what we are going to do is now to write down the simplex method in a more formal way; this is given from (( )) Padberg's version on his book, go back to the black one. So, what is step 0 is to initialize everything; find a feasible basis, its index set I, that is you have to know what are the one what are the columns, which are inside the basis matrix; and initialize p k for all k in I. So, basically you which p k is the which original column of the original matrix say is now in the basis matrix; if none exists and no basis feasible basis exits, feasible basis matrix, I should write feasible basis matrix if you want to (( )) more in detail, then stop, LP is increasable, there is no feasible basis matrix, so there is no feasibility of the problem itself.

Else, if you can find, compute B inverse and b bar equal to B inverse b, and initialize C B which corresponds to this b, then you know what is what it means; step 1: Compute C bar of course, A is given into two parts B and n is the n part which is important, but does not matter in such do this computations; if C bar is greater than equal to 0, then set x B equal to b bar and x n part or the reminder part whatever whichever sign you want to give the non-basic part to be 0, and this is the optimal, that is x equal to x b 0, is optimal.

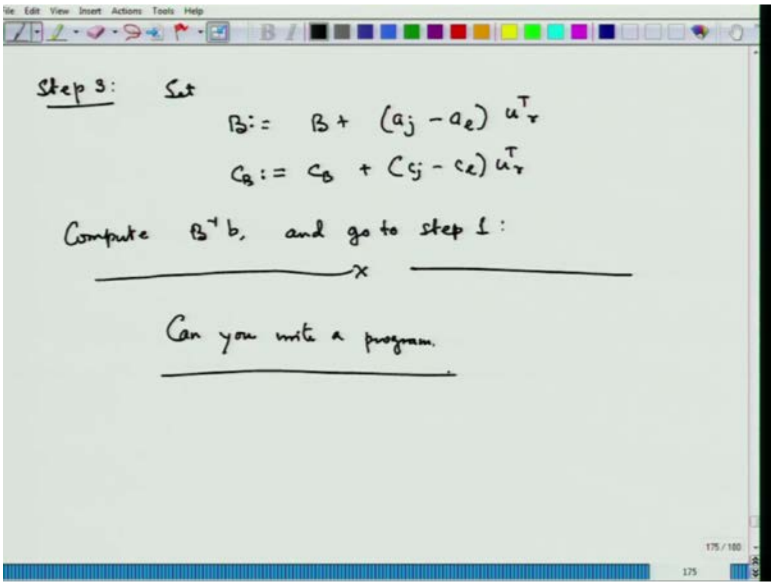
(Refer Slide Time: 44:27)



Else, if this does not happen, choose j element of all k, which belongs to J minus I, but for which C bar k is strictly lesser; choose one j, basically you choose the that value of j, for which among all the case C bar k is of the least value so, that you choose as j. So, everywhere we have to put end of the loop, which I should be putting. So, here also I should put a, this is if none exists then this end loop has to end end if if loop ends, here if this happens or else choose j and end the loop; step 2: Compute this is also very important to know, we have chosen the j what is this; now if, so it is all if then loop, if then else loop basically, then stop LP is unbounded, this is already we have proved; else sorry if this does not happen, then compute theta, which is mean of you know, how to do it. So, this is exactly what we have just learnt.

Choose the l eth position in I, such that b bar p l that is p r is l basically in our case r is p l. So, k r is equal to p l. So, p l is that l eth position p l is l, when a p k r is r. So, set r is equal to p l, so this is the… So, the from the r eth position, the vector in the r eth position a r the a l would actually get out, that is exactly what we have done nothing else. So, again we end the if loop, so you put even think of writing a program on this.

(Refer Slide Time: 47:31)



So, now the iterations the updating step 3 are set is B dash sorry B is now assigned to, I cannot write it in that format, I am writing in a computer programming format, so C B, so this is assigned to basically, you can also give an arrow like this, it does not matter; compute B inverse b and go… So, with a new b you compute B inverse b and go to step 1 and do the same checking all over and again. So, can you write a computer program with this? That is very important; can you or can you write a program that will be fun to do this. So today, we went the simplex method; and in the next class, we are going to speak about a very exciting topic called interior point methods in linear programming; these whole ideas of interior point methods, which give polynomial type algorithms, actually revolutionize the field of convex optimization. And so, what we are going to do from the next class is a very, very interesting and some sort of a regulation topic in optimization theory; thank you very much.