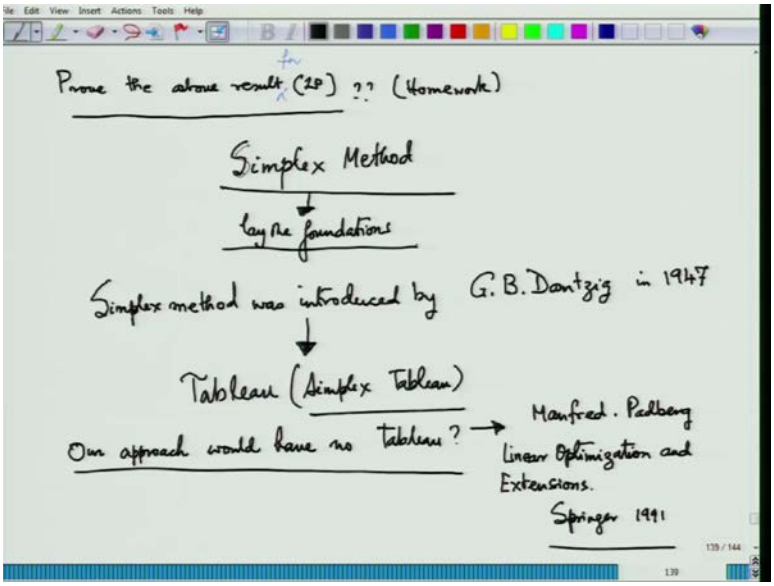
Convex Optimization
Prof. Joydeep Dutta
Department of Mathematics and Statistics
Indian Institute of Technology, Kanpur

Module No. # 01
Lecture No. # 23

So, in the last class you remember what we did was to prove that if a linear programming problem is bounded below, then and I took a linear programming problem of a certain more relaxed kind and showed that if it is bounded below, it has a minimum a minimize are exist. Now what we were taking this part of the convex optimization course is on this special topic of linear programming, a special convex optimization problem, which I would like to call this little part of the course - a sub course, which is called the pleasures of linear programming; because possibly this is what one can call as real mathematics; beautiful mathematics, beautiful results, beautiful algorithms and at the same time, beautiful applications and huge applications.

Now if I go back to what I wrote in the last class, meanwhile I took a little bit of you might say vacation or little bit of holiday. So, like you may saw I am coming back after the somewhere I am talking to you and so, here I have written prove the above result for LP, actually the last result, which I requested you to prove for LP means our LP, the standard form.

(Refer Slide Time: 01:36)



So, you should we for does not matter, so just did the correction with the blue one. So, here again, I would now go over to the simple method, so simplex method. So, what I am going to do is to lay the foundations to this method; (No audio from 02:05 to 02:17) simplex method (No audio from 2:19 to 2:26) was introduced rather developed by the famous operation researcher or actually a mathematician George. B. Dantzig in 1947. In fact, it is good to tell you how the name linear programming came, because once George Danzig was taking a walk with a very famous economist T. C. Koopmans; and Koopmans asked him George what are you working on right now, he said there I am working on certain problems, which has to do with minimizing linear functions subject to linear constraints that are fine constraints.

And these problems are coming out of applications for certain programs of the US navy, certain strategical programs for the US navy, in order to solve those issues; so, but I do not know what should I called issue; should I call it optimization with linear of objectives and constraints? Koopmans said, why do not you call it linear programming; and that is why the name linear programming as well as mathematical programming, which largely talks about any non linear convex optimization problems came into vogue.

Now simplex method is usually thought in standard under graduate classes, using what is called the simplex tableau - a table; tableau or simplex tableau. So, it is tricky way to solve the gauss elimination process and also at the same time, do some book keeping to keep up the check on optimality; now let me tell you that I will not discuss the tableau. Our approach would have no tableau; (No audio from 04:30 to 04:39) those who have already done some linear programming and possibly was in this course to talk about about something about convex programming; they can be rest assured that I would not board them again with the tableau. So, you might be asking and for those who have not, I would also refer to you a course by in the same NPTEL programme, which has, which talks about tableau methods these are the course completely on linear programming.

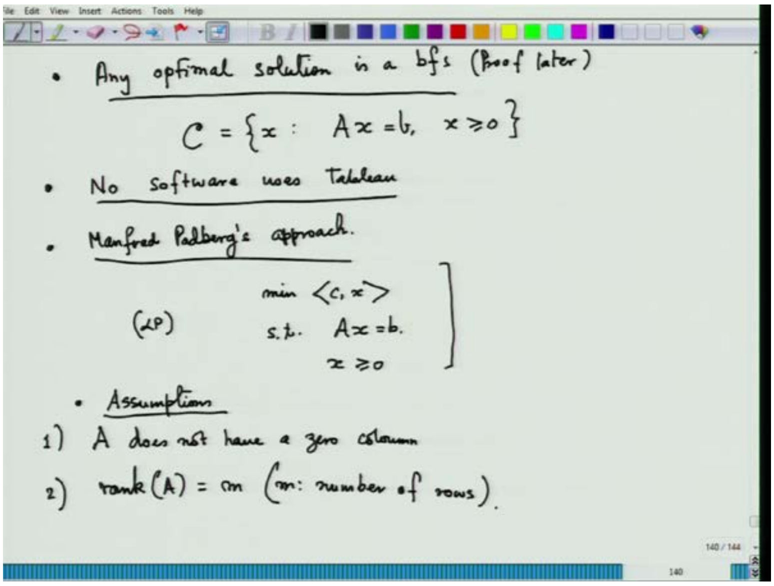
So, why our approach would have no tableau; this tableau less approach was possibly pioneered in the book by Manfred Padberg who is in (( )) for the student of Danzig. So, he wrote a book called linear optimization and extensions, (No audio from 05:28 to 05:43) it was published by Springer, I think way back in 90 or 92, something like that; I guess 91. So, this book, he promotes the idea of doing linear programming without the tableau method; you might ask how that is possible, for those who have already done some linear programming.

First of all what do you do in optimization, when you run an algorithm; you take a start or test point x naught - initial guess point; because you need to start the algorithm, so, you guess something; is this the solution, you make a guess. So, there will be two options; you say yes it is, it is really a solution, then you stop the algorithm; and if you say no, if is not, then by some approach, you move from x naught to x 1, such that the functional value, say if the function my objective function is f, the functional value at x 1 must be strictly less than f of x naught provided that we are minimizing the function.

So, you come from x naught to x 1, where this property is. So, you make a descent in terms of the functional value. So, this direction wherever which you move from x naught to x one would be call it direction of descent; now this is the basic (( )) fundamentals of an optimization algorithm; how you go from x naught to x 1 is the question, which would interest us, because there are many, many ways to go through and lot of research still goes on how to go from x naught to x 1 for various types of optimization problem.

But when you look at the tableau, the simple method; you see there does not tell me, it immediately it is not apparent, whether it is telling to go from x naught to x 1; it is telling me to go from x naught to x 1, because that is what any optimization algorithm would do, but just by looking at it is not possible to tell you whether it is going from x naught to x 1, but here we make a clear picture that we are going from x naught to x 1 and it will be done.

(Refer Slide Time: 08:14)



So, a very important thing to keep in mind when you do a linear programming is that any optimal solution is a basic feasible solution; any optimal solution of the linear programming problem is a basic feasible solution; any optimal solution is a basic feasible solution or bfs. Now, you might ask me how I accept your things; those who already know convex optimization of just looking at this video for fun to see what is up there in the net. So, you would immediately realize that you can convert a linear programming problem into convex maximization problem, and then you know the it will be on the boundary of the solution; and not only in the boundary, it will be one of the vertices, if the thing is polyhedral. So, let me not get into those details, but I will prove this fact, but for the time being you consider this as the fact, which is the true fact. So, we will prove it; proof later.

So, what does simplex method will do, simplex method will take a bfs and go from one another bfs to the next bfs, because every vertex of the convex polyhedral, which is the feasible set that is the if you look at the convex polyheda C or S, I guess C, I was marking as C or particular; if you look at this, this is a polyhedron and this as vertices. So, every vertex of the polyhedral C, we we know is a basic feasible solution, but then if the number of vertices are very large, which happens in most problem, you cannot keep on computing each and every vertex, and then trying to enlist from in a ascending order, so you know what the minimum.

So, this is the very time taking process and would be an NP hard process, if they are allowed in this be too large. So, if the number of vertices could be too large. So, what would you do? So, we have to find a clever way; the clever way is that if I have a bfs now, which does not correspond to the solution, then I move to another bfs, another vertex in such a clever way, such that in the new bfs, when I get the new bfs, the basic feasible solution or the optimal value corresponding to the or the objective function corresponding to that particular bfs would be strictly lower than that the the objective value for the previous bfs. So, I have make a made of descent.

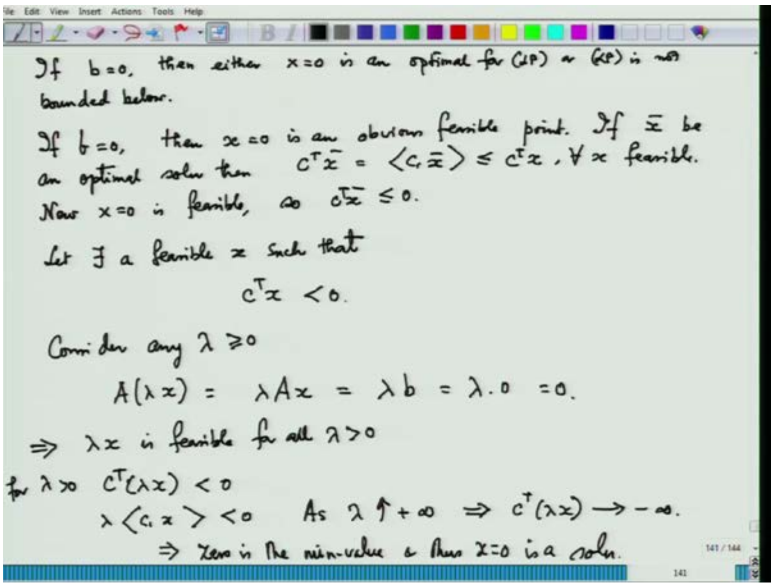
So, the simplex for the does exactly the same thing; we will also now do the simple method, but without the tableau. So, the tableau, what we do is a simple method; and the simple method lying any optimization method goes from x naught to x 1, but if you look at the tableau, it is not clear whether it is going from x naught to x 1, because you get (( )) down into a series of calculations, which is same as the pivoting technique in Gaussian elimination; and you get (( )) down with those calculation and the real idea behind the simple method goes off; and those who would read Manfred Padberg, a very (( )), but if you read it, it is possible you are one of the best mathematics books, I have read.

So, if you read Manfred Padberg, you would come across this fact that no software uses tableau; this tableau as you see can be done for certain 6, 7 variable problems possibly with hand, but no software which are supposed to deal with the large and large amount of, large amount of variables, a huge large linear programming problems, would really not deal with this. Now the question would be then if it is not dealing with this, then how does the software actually compute, how does the software do the simple method?

So, we will take the approach, which will tell you how the software actually would run the simple method; our approach would based on Manfred Padberg. So, we are using Manfred Padberg's approach; there will be no tableau; and that with this approach I am giving is from the book; I am sure many of you would not have read it or even heard of this book; what a wonderful book; not only from the point of view of optimization, but also mathematically exciting. So, again we now go back start our foundations of the simple method. So, our problem again to recall those who do not remember what has been done earlier is to really minimize this linear function, such that A x equal to B and x greater than 0.

So, in this approach Manfred takes certain assumptions, which are quiet natural. (No audio from 13:30 to 13:41) So, the assumption is that A does not have a zero-column; so see, if it is a it has a 0 column, so there is basically one particular variable does not come in the feasible set. So, that variable can take any value that you that you can give. So, you can really fix up the other variables, which are actually appearing and then put whatever value you want for that variable. So, so A if have a zero-column, that zero-column would not interest us. So, for a computation point of view, we do not want any which has a 0 column; if it does not have a zero-column, its true if it is a zero-column, we really have to get rid of that and try to do the thing. So, the first assumption is A does not have a zero-column and second which is a standard assumption that is a full row rank. So, rank of A is m, where m as you know is number of rows.

(Refer Slide Time: 15:05)



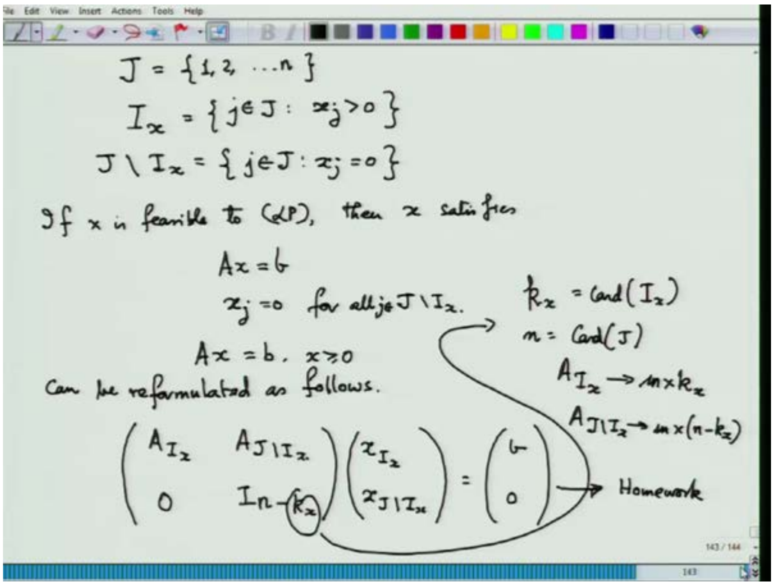
Now, let me state you curious fact; what about it, do you have any assumption on B? The answer is interesting, because if b is equal to 0, then either x equal to 0 is an optimal for the linear programming problem or LP is not founded below. So, here there is a quiet little proof in Manfred padberg, which I want to produce. So, if b is equal to 0, then x is equal to 0 is an obvious feasible point. Now if x bar be an optimal solution, which we assume it exist, then C transpose x bar or C of x bar is greater than equal to C transpose x; sorry, sorry I made A mistake C transpose x bar, which is a solution is less than equal to C transpose x, for all x feasible.

Now x equal to 0 is feasible. So, C transpose x bar is less than or equal to 0. What we have to show that that C transpose x bar less than equal to 0 is strictly less than equal to 0 is not possible; we must have C transpose x bar equal to 0; if C transpose x bar is strictly less than 0, we will show that the problem would be unbounded below. So, if C transpose x bar is equal to 0, then of course, x equal to 0 is the solution, is a solution, not the only solution, but possibly is what, is a solution.

So, let there exist a feasible x, such that C transpose x is strictly less than 0. Now, consider any lambda greater than equal to 0, may be greater than 0 does not matter. So, A of lambda x is lambda A x is equal to lambda of b and b is equal to 0, so lambda into 0 is equal to 0. So, this would imply that lambda x is feasible for all lambda greater than 0; therefore, C transpose lambda x obviously, because if I take lambda greater than equal to say say for lambda greater than 0, C transpose lambda x is also strictly less than 0. So, which means lambda transpose C x is strictly less than 0. So, as lambda tends to plus infinity, because C transpose, see this C x is less strictly less than 0, it implies to sorry not go down, goes up to plus infinity; lambda goes up to plus infinity, it is clear that this one that is C transpose lambda x is immediately rushing to minus infinity.

So, which proves that this unbounded below, if C transpose x, there is a feasible x as the C transpose x is less than equal to 0. So, when b is equal to 0 C transpose x bar cannot be strictly less than 0, but it has to be equal to 0. So, if it is equal to 0, which means C transpose 0 is also equal to 0, which is the minimum. So, 0 is the minimum value and so, which would imply that x equal to 0; so 0 is the min Val - minimum value and thus x equal to 0 is a solution. So, in most cases you will not see in that real problem that b is equal to 0, b is hardly equal to 0. So, to begin with let us fix up some notations; notations are very important as we go along, so we will fix up some notations.

(Refer Slide Time: 20:01)



Let us consider J to be a set of indexes marking the rows of the matrix m cross n matrix a full rank; an I x for a given x, I should write I of x, but I of I Just writing it like this, to keep it separate from the active index at that we have already spoken about while studying Kuhn-Tucker conditions; keeping it slightly separate than the active index set, because here it is not exactly the active index set, it is essentially some sort of inactive index set, which would be important for some time.

And J set minus I x, would consist of all J in J, such that x of J is equal to 0. So, for a fixed x, we are just collecting indexes, these are indexes. So, if x is feasible to LP, then x satisfies that we can write down the fact that x – A x is equal to b and x greater than equal to 0 as follows, x equal to b and x of j is equal to 0, for all J, for all J, which is in J, but not in I x. So, we can write the matrix, it is similar to the splitting that we did for basic and non basic is quiet similar, but if you write the matrix now of the equation we can write it in this way.

So, A x equal to b can be reformulated; (No audio from 22:00 to 22:18) so consider A I x that is this is the matrix, whose rows consists of those rows. So, your column is same, it consists of those rows, which belong to I x, J is 1 two m. So, it consists of those rows, which belong to I x. So, we can write down this form in this partition form; we will write down what everything means. So, A has m rows and J has m rows what is A I x? So, how many columns this matrix has; we will all fix it up right now, where we has a slight

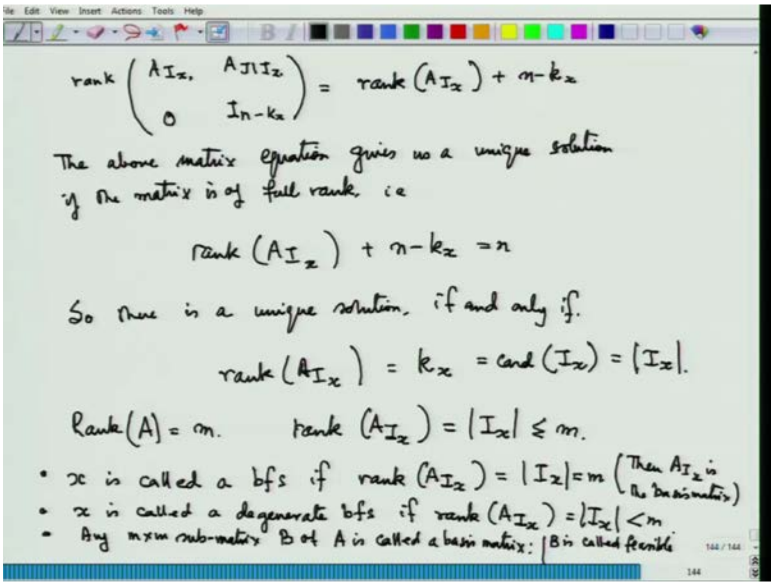
(( )), we Just hold on with me for some time and I will just… (No audio from 23:19 to 23:28) Now you will see what is this? What is this K x? So, this is something, we need to know.

So, let us write down k x is cardinality of I x, some cardinality of this set. So, this is the cardinality of I x right. So, corresponding to the cardinality of I x, I have split the A; suppose I x is say this one; sorry not m, I make here mistake; this should be n; please take a note. So, you have this whole x J vector, when x 1 x 2 dot, dot, dot n, x n in R N and now you are splitting it up into two parts; first part is you are taking some indexes for which x J is strictly greater than 0, some index which is x J is equal to 0, because x J must be greater than or equal to 0 each of exist.

So, we take those indexes, for which those particular columns, for which x J is strictly greater than 0; we assumed that they are in the first part. So, we take those separate and remaining n minus k x right. So, n is obviously, the cardinality of J, which is the counting of the number of components your x is in R N. So, remaining J minus I x is the case, where x J equal to 0. So, you split the n columns into these belongings; some columns for corresponding to which x J is strictly greater than or some column for which x J is equal to 0. So, now it is not that the first I x column, so first k x columns are having x J greater than equal to 0, it could be otherwise, but just by applying a permutation matrix both sides, you can make this arrangement.

So, finally, what we get is a similar matrix of this type. So, from the point of your solution, this matrix and the one if you had not kept A i x all on one side, but something in the some of them in the middle, then also it would give me the same solution. So, A I x has is of the order m cross k x. So, A J minus I x is of the order of m cross n minus k x. So, x I x, if you are multiplying, if you take this A, so there am I x rows, so you can multiply with this I x column, so you multiply with this I x rows here, so these are strictly bigger than 0. Remaining part J minus I x, all of them x Js are equal to 0. So, you can see that what you finally get is exactly and here this is 0, and this is the identity matrix of x J greater than x J is equal to 0. So, if you look at this, you will exactly get back this. So, your homework would be to check that this corresponds to the solution of, corresponds to this equation; so homework is to check this and this.

(Refer Slide Time: 27:12)



Now once we know this notation, we have the following that the rank of A I x A J minus I x 0 I n minus K x, this thing is can be is same as rank of… (No audio from 27:29 to 27:37) Why, because m minus K x corresponds to linearly independent columns and only the linearly independent columns in this matrix would now add up to give you the rank of away, so we have spitted up like this. So, the above matrix equation means the one, which is in the previous page, gives us a unique solution, if column rank equals row is equal to the number of, is the column rank or row rank, of the row rank is equal to the number of columns. So, it is a full rank.

That is if the matrix is of full rank, (No audio from 28:32 to 28:39) that is rank (A I x) is rank of this matrix thus n minus K x is equal to n. So, there is a unique solution, if and only if rank of we just do certain simple algebra to the cardinality of I x or cardinality of I x is also denoted like this, which we know. So, this is something we can immediately come across, so since rank of A; so, A is basically participated partitioned in these two parts. So, rank of A is m; m is full rank. So, rank of A I x sorry rank of A I x would also be m of course, look like to be m, but that is not the case. So, rank of A I x, which is I x that is if it if it has a full solution; rank of A I x, if you look at it; rank of A I x, here I do not have all the n columns; I have the number of linearly independent columns in this matrix has to be less than m, because rank of A is m. So, in the matrix A, there are only m linearly independent column. So, if I take all those rows and only few columns, then what I am getting is rank of A I x must be less than equal to the total given rank rank m.

So now, we will have certain… In order to study the simple method, we will certain takes makes certain notations, which is the Manfred's (( )) style of notation. So, we will maintain that sort of notation in our study of the simple method. So, we know what we called bfs by partitioning; b n and n all those things; here also do I doing the same thing by partitioning, but giving a different name to a different symbols to it; we will say x is called a bfs, (No audio from 31:08 to 31:14) if rank A I x. So, your solution would be a bfs, if this is exactly your b, b part, which which will be a square matrix.

So, rank of A I x would be exactly equal to I x, if m is more than m. So, m should be here, it is its full rank; so, rank of A I x should be is equal to I x. So, if they that is what we will call a bfs; in the sense that, if this rank A I X is full rank and the number of columns in this is also m. So, this is exactly would be your b, that we have seen that b and n separation in the earlier lectures, that is the exactly this story that these story is written like this. So, if this rank of A I x is m, which is rank of A I x is I x and which is m, then we call it a bfs. So, what I want? I want the rank of A I x should be equal to I x. So, rank of A I x is equal to I x, then we have a unique solution to this problem.

So, this, so there is a unique solution to the, this linear equation, it has a unique solution. So, then rank of A I x would be I x, but if this I x is equal to m, then this is your basis matrix; A I x becomes, then A I x becomes basis matrix; A I x is the starting basis or just basis is the basis matrix. So, what we have done? We have written, we have taken a j, which is a, which is what we have done let us recollect; it might not be so easy for everyone. What we have done so far?

What we have done so far is that if x is a feasible solution, then x satisfies these equations right. And we are telling that if x is the only solution of this equation right, that is having these, these particularities, that x is equal to 0 for this number of for for all this Js; then then A x equal to B and x greater than is equal to 0; or rather I would say this fact, this expression can be reformulated, this can be reformulated as this whole expression can be now reformulated as follows means, writing this is same as writing this, and which I say, it is of this form; and if this there is only one x, which has those properties with their x Js, that is ok; I have an this is the only possible x, for which x is partitioned like this and is solving the system; then rank of A I x must be equal to the cardinality of I x, and if the cardinality of I x is also equal to m.

So, then x is the only solution of that particular system or equation which we just show. And then that solution is called a bfs, if its rank that is cardinality of I x is also m. So, its slightly different way of telling the same thing that if this happens, then this is your actual B. So, x is called a degenerate bfs, (No audio from 35:09 to 35:24) is rank of A I x is equal to I x is equal to m, maybe I should write it more clearly. So, any m cross m sub matrix B of a is called the basis matrix; and x is called a bfs, a basis is called feasible sorry x is a bfs, it is already given; B is called feasible, because you have seen that you you when you compute the basic feasible solution, there is two parts x B and x N; x N is 0 for a basic feasible solution and x B is nothing but B inverse B of some such a sub matrix, such a basis matrix. So, B is called feasible, so if x, it has to be a bfs, then B inverse B has to be 0; then B inverse B must be sorry not have to be 0, has to be greater than equal to 0. So, these are the basic notations that we require.

So, once we know the notation, we need to talk about some very, very fundamental stuff about linear programming. So, the main result that we are going to prove that if there exist x element of C, then there exist a bfs x bar element of C. So, if x is feasible, we have a feasible solution at it is a polyhedron, then there is a bfs; and if there is an optimal solution, there is an optimal bfs also, basic feasible solution. This is the most important result that tells you that an every bfs we have proved to be a vertex, so there is a one to one correspondence, so every basic optimal, optimal basic feasible solution lies at the vertex. So, we have to jump from one vertex to the other, but do it in a clever way. So, that the objective value decreases strictly; and that is exactly what is the simple method going to do.

And that clever way is that technique, simplex technique, but it is somehow important that we need to not only know this results, but from mathematical point of view, we need to work down through the proofs of this result, because this in the proof of this result, the inherent technique of simplex method is hiding, in the proof of this result; and then we will when we will write down the simple method you will soon see what I am telling is so correct.

So, so, tomorrow we will concentrate on proving two results; the first result is if x is the basic feasible solution, and x has at most m positive components, because rank of A is m; and the sub matrix, which we have defined is A I x, this can be extended to a feasible basis, but adjoining A I x with some more columns from the remaining part; if this is not m, number of columns I can take something from this and make it into a basis matrix and only a basis matrix it can be made into a feasible basis matrix; and then once that is done, we have to prove the very fundamental theorem of linear programming, if there exist an x element of C, then there exists a bfs; if there is an optimal solution to Lp, then there is an optimal basic feasible solution; this is the exactly what we will study in tomorrow's class .