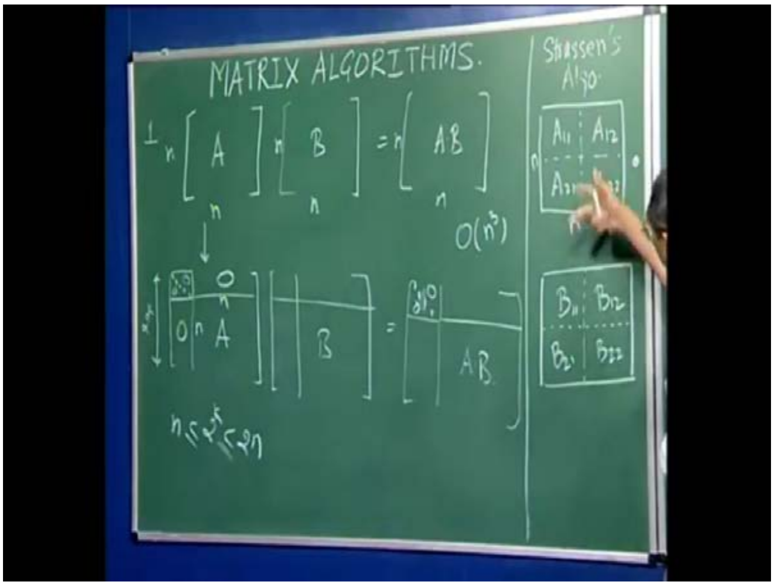
Computer Algorithms II
Prof. Dr. Shashank K Mehta
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Lecture - 12
Matrix Inversion

Hello, today we begin the discussion of matrix Algorithm, we will start with matrix multiplication and then we will give some algorithms for inversion and decomposition.

(Refer Slide Time: 00:36)

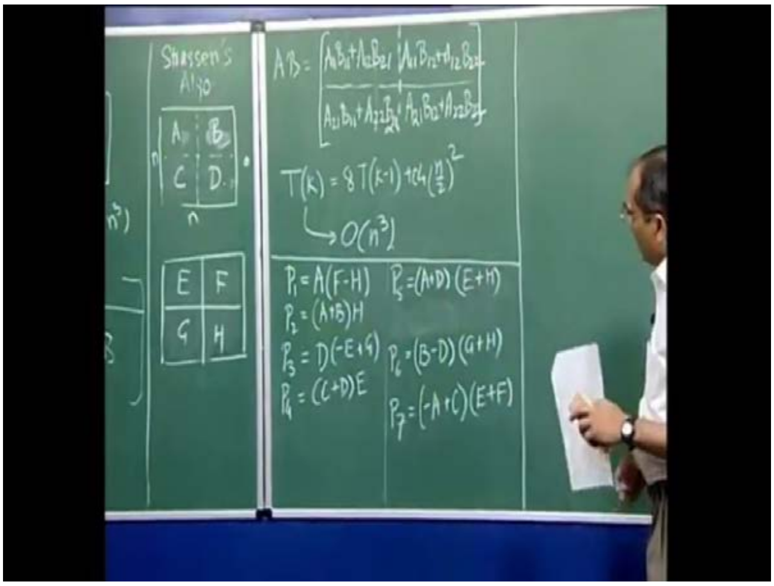


Now, first thing I would like to show that suppose we have a matrix of size n cross n, say this is matrix A and we would like to multiply this to another matrix of the same size, then indeed the result is also of the same size. Now, often times out that it is easier to design algorithms, when n is a power of 2, now in that case suppose we embed such a matrix in a larger matrix, we have some size 2 power k greater than equal to n. This is an n cross n matrix A and this is another square matrix of size 2 power k minus n by 2 power k minus n. We put a unit matrix here, we put all 0s and we do the same with the other, then the result will be again the unit matrix.

So, maybe I will show that by one and this will be the product of the two matrices A and B, so what we notice is that we can embed a smaller matrix in a larger one by putting identity matrix in the orthogonal dimension, the other dimensions. Then, those involved in a then those end up giving us one the unit matrix into unit matrix as unit matrix and the result of A and B comes here. Hence, one can always extract the result of A times from this part, hence there is no disadvantage in assuming that this is a power of 2 dimension indeed. Of course, the cost can go up because this is a higher dimension, but one thing you may want to notice is that this dimension this power of 2 has to be between n and 2 n.

Hence, at most size can double, but not more than that now we will describe, now in fact first thing we know is that the that normal longhand algorithm for multiplying two matrix is takes the n the n component vector the row vector. It takes the n component column vector takes the dot product and puts here and so on. So, what we have here is that each of the component of the product results by computing is the dot product of two vectors of price n each competition requires n operation n multiplication n minus 1 additions. So, this is an order n toss, hence the entire matrix takes n square into order n, therefore order n cube algorithm is right away known to us.

(Refer Slide Time: 06:25)



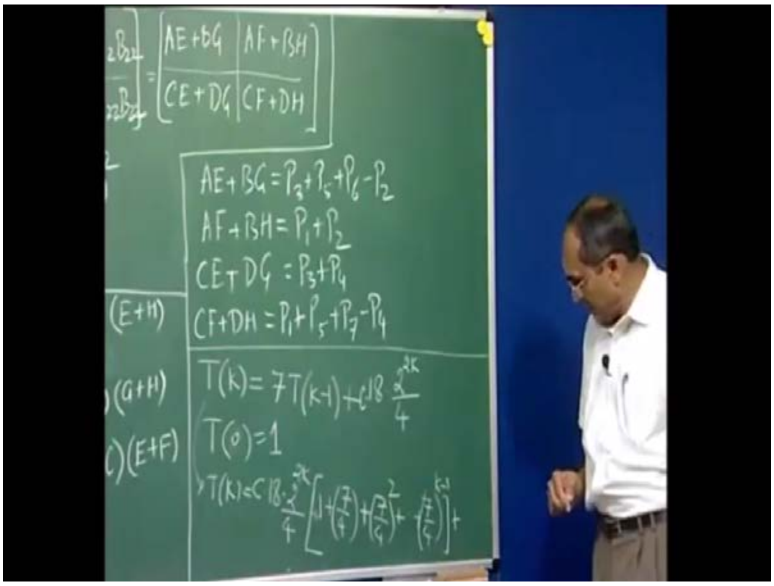
I am going to describe C slightly efficient algorithm due to Strossen’s algorithm by a skilful manipulation of these structures. From now onwards, I am going to assume my n is to power k because one can always perform this manipulation and assume this is your n. So, suppose we have two matrices which are of size n by n if I split them into four half size matrices, I am going to call them A 1 1, A 1 2, A 2 1 and A 2 2 and I am going to multiply this with matrix B. This is also split into sub matrices B 1 1, B 1 2, B 2 1 and B 2 2, now in fact lets us express the product of these two in terms of some matrices.

So, A dot B will this block multiplies to this block and gives you A 1 1, B 1 1 plus A 1 2, B 2 1, B 2 1, this block will be A 1 1, B 1 2 plus A 1 2, B 2 2, B's will be A 2 1, B 1 1 plus A 2 2, B 2 1 and this is A 2 1, B 1 2. This A 2 2, B 2 2 involves 1, 2, 3, 4, 5, 6, 7, 8 matrix multiplications of size n by 2 by n by 2 and one can hence express the total cost of multiplying two 2 to the power k matrices, this denotes the log of n 2 to power k by 2 power k matrices. It involves eight multiplications of half size, in addition to that we are going to perform certain summations and the total number of submissions will be n by 2 square four of them.

So, we have 4 n by 2 square, let us say we put a constant here C times this and one can solve this recurrence relation and we can easily see that this terms out to be order n cube. Hence, it gains nothing by just taking this approach this is a divided conquer approach, but does not lead as to any better result than what we already have. So, Strossen's proposes a trick by which he reduces the total cost and that is we trick that we have to now look. What we do is we define the following matrices as A F minus H P 2 A plus B times H P 3 D minus E plus G P 4 is C plus D times E.

In addition to that, we also define three borne matrices P 5 is A plus D into E plus H, this is P 6 is B minus D G plus H and P 7 minus A plus C and this is E plus F. So, let us consider these seven matrices since I have changed the names I will correct them, these are A, B, C, D and these are E, F, G, H. Then, it is a matter of just simple verification one can show that A 1 1, now these are now I have to modify them.

(Refer Slide Time: 12:28)



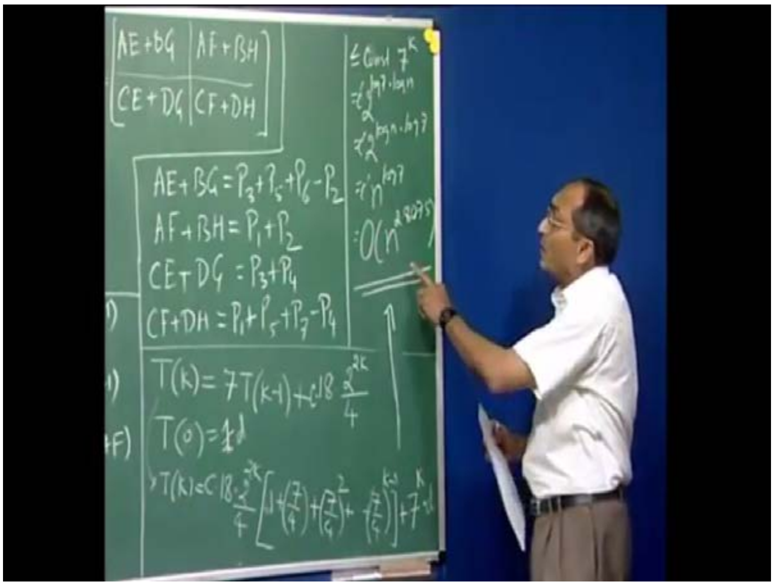
So, that becomes A E plus B G A F plus B H C E plus D G and this is C F plus D H, now we can express these matrices in terms P in the following fashion. We can verify that A E plus B G is P 3 plus P 5 plus P 6 minus P 2 A F plus B H is P 1 plus P 2 C E plus D G that is this matrix is P 3 plus P 4 and lastly C F plus D H is P 1 plus P 5 P 7 minus P 4. Now, what you have to observe is that these are all n by 2 by n by 2 matrices A as well as F minus H a plus B H all of these are half size n by 2 cross n by 2 and what we have then here is we have computed seven products and from that we can expect all the four sub matrices of the product.

This is our A times B this is unlike the direct approach, when we at computed 1, 2, 3, 4, 5, 6, 7, 8 product, this is the essence of the Strossen's algorithm and that leads to the saving which is we see here turning out to be n to the power 7. So, let us this write down the recurrence relation t k t k is the product of 2 n cross n matrices k is log of n.

This involves multiplication of seven matrices of half this size, in addition we have added several half size matrices and those include 1 here, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 additions of matrices of half this size. So, it involves 18 into each addition is n by 2 square, so that is 2 to the power 2 k over 4 that many. So, again I will just put some constant here these many operations be performed of additions and the end of with our matrix. Now, at the base case what is going to happen in the base case when we break up it 2 cross 2 matrix, we are going to generate 1 cross 1 matrices and the cost of multiplying a matrices of my mistake.

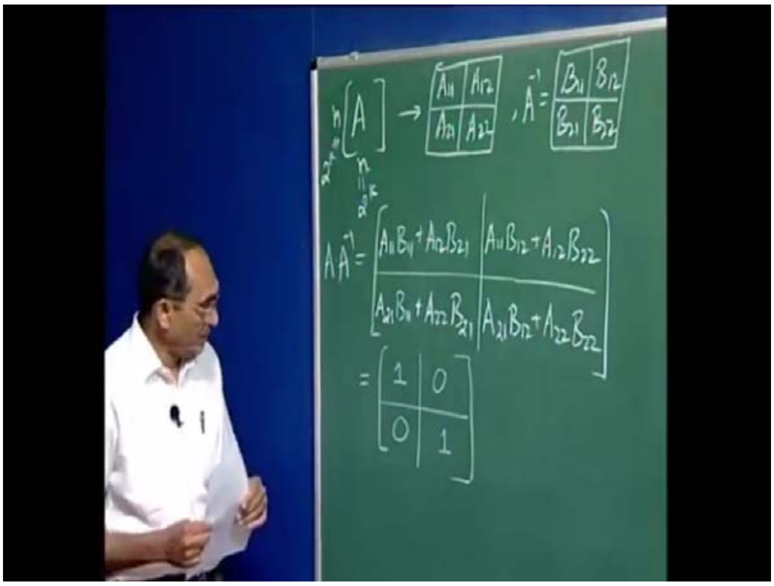
It should be 0 that is of dimension 1 cross 1, so will be because we are writing log here, this will be one or a constant order one constant here. So, let us just solve this equation this gives me t k equal to some constant times 18 times 2 power 2 k by 4, next time you will have seven occurrences. So, seven of these with half the or one-fourth size because we are squaring it, so we will have 1 plus 7 over 4 plus 7 over 4 square plus 7 over 4 k minus 1. At the end we will have only 1 cross 1 size matrices and they will be in all how many, so we will have seven of this 7 square of t k minus 2 7 cube of t k minus 3. At the end, we will have 7 to the power k of t 1, but t 1 I can just write down as another constant say D high could simply say this is D.

(Refer Slide Time: 19:24)



You add this terms out to be seven some constant, this is less than equal to 7 to the power k which is equal to 2 to the power log of 7 into k is log of n, which is 2 to the power log of n times log of 7, which is n to the power log 7. Again, it is constant here, which is order n to the power 2.8075, which is again over n or n cube algorithm, now we are going to discuss a how to compute the inverse of a matrix.

(Refer Slide Time: 20:33)

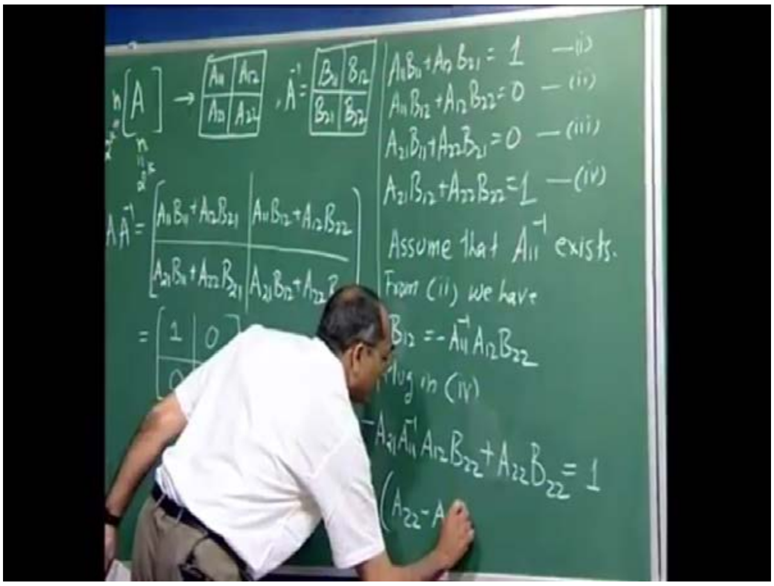


So, let us suppose we have a matrix A which is half size n cross n, now you know that you cannot compute the inverse of a matrix unless it is non singular, you that now at this stage we are not going to compute a going to give a exact algorithm. We have performing certain symbolic computations, so what we are going to do is we assume that a which is assuming that n is of course a power of 2, we can break it this into two parts into four parts A 1 1, A 1 2, A 2 1 and A 2 2.

Let us suppose this matrix is invertible and let us say a inverse is a matrix which also can be split into four parts and this is B 1 1 B 1 2 B 2 1 and B 2 2 well by definition the product should be the identity matrix. So, let us write down the product and we are going to get A 1 1, B 1 1 plus A 1 2, B 2 1, A 1 1, B 1 2 plus A 1 2, B 2 2, A 2 1 plus A 2 2, A 2 1, B 1 2 plus A 2 2, B 2 2. This would be the product, which should be should be equal to D identity matrix, which I can also view in this split form.

So, this will be an n by 2 cross n by 2 identity matrix, this will be also an identity matrix this will be zero matrix, so will be this one. So, now we can equate these two identity matrix, these two identity matrix these 2 to 0 to get the four equations as follows.

(Refer Slide Time: 23:37)

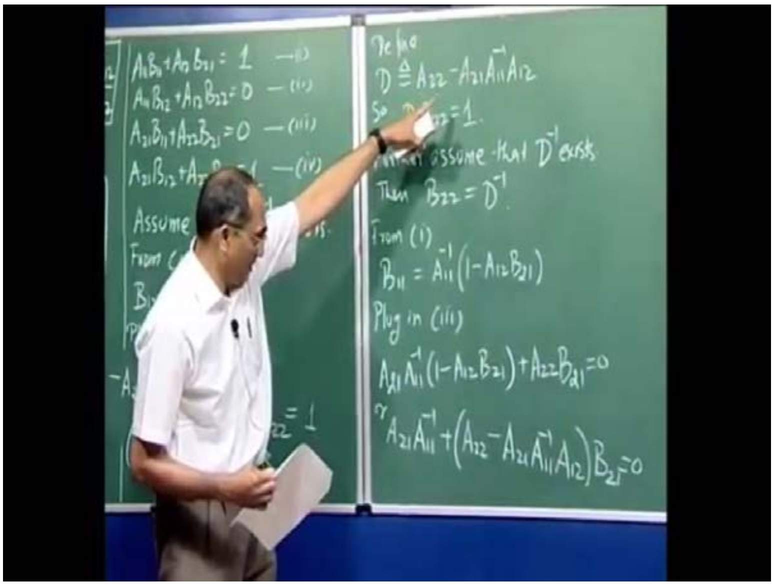


You have A 1 1, B 1 1 plus A 1 2, B 2 1 equal to 1 this one stands for a unit matrix of size n by 2 cross n by 2. Then, we have A 1 1 B 1 2 plus A 1 2 B 2 2 is should be that zero matrix the matrix is all 0 entries in size n by 2 cross n by 2 A 2 1, B 1 1 plus A 2 2, B 2 1 is 0. Finally, A 2 1, B 1 2 plus 2 B 2 2 is 1, now we would like to simplify them, but for that I need to make one exemption. Let me assume that matrix A 1 1 is self is invertible please note that such a thing need not in general be true the matrix A maybe invertible. That does not imply this matrix is itself non singular, but let us assume that A 1 1 inverse exists.

We will now solve from this equation, we will get an expression for B, so let me just label them as 1, 2, 3 and 4, so from two we have A 1 2 equal to, sorry B 1 2 equal to minus A 1 1 inverse A 1 2, B 2 2. What I have done is a taken this product on the other side becomes minus A 1 2 times B 2 2 and then multiplied by A 1 1 inverse from the left side. So, that cancels of leaves be one two on the left hand side and we get minus A 1 1 inverse A 1 2 B 2 2.

Then, let me plug in B 1 2 in four plug in equation 4, you are going to get minus A 1 1 inverse A 1 1 to B 2 2, sorry so that could be minus this was B 1 2 minus A 2 1 followed by this expression. Then, we have plus A 2 2 B two equal 2 1, so let us rewrite this is A 2 2 minus A 2 2, A 1 1 inverse A 1 2, B 2 2 equal to 1.

(Refer Slide Time: 28:16)

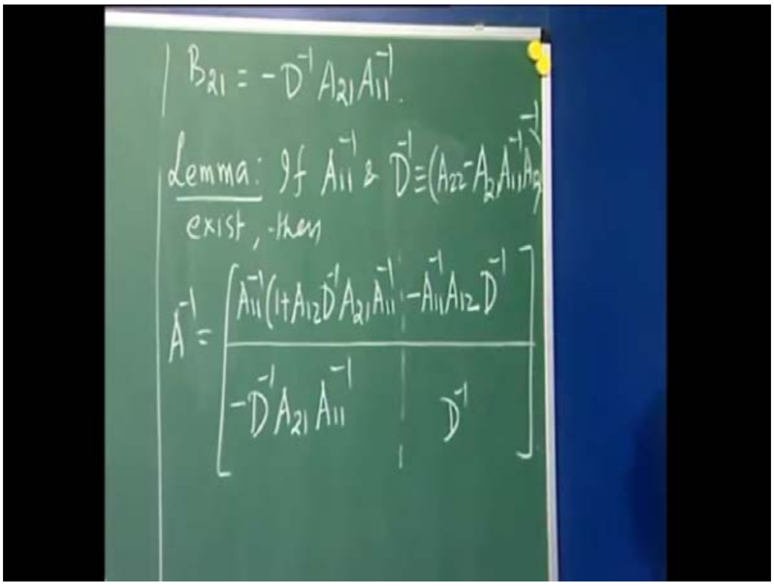


Now, let me define a matrix D this denotes I am defining D E to be A 2 2 minus A 2 1 A 1 1 inverse A 1 2. Now, let me sure I am doing yes in that case, so this equation is simply D times B 2 2 equal to 1. Now, let me make one more assumption further assume that D inverse exists once again such a thing need not be in general true, but suppose it does its exists, then B 2 2 is D inverse that is simply multiplied by D inverse from left on both sides to get this. Now, we can logging the value of B 2 2, let me see we have B 1 2 we also have B 2 2 and we still have to find out B 2 1 and 1 1.

So, let us take the first equation, now from one I have B 1 1 notice that A 1 1 inverse exists. So, I can multiply this A 1 1 inverse from left and that is going to give me A 1 1 inverse one minus A 1 2, D 2 1, D 1 1 is now expressed, sorry B 1 1 is expressed in terms of D 1 1 inverse A 1 2 and D 2 1. Let us plug this B 1 1 in two equation tree, so plug it in equation 3 and that leads to A 2 1 for B 1 1, I can write down A 1 1 inverse one minus A 1 1 to B 2 1 followed by A 2 2, B 2 1.

This is a consequence of plugging in the value B 1 1 in equation three, now we have B 2 1 in both of these, we need to get an expression for B 2 1. So, let us collect them together I am going to get A 2 1, A 1 1 inverse plus A 2 2, so the next term is this times this and this. So, I am first writing down this 1 minus A 2 1, A 1 1 inverse A 1 2 times B 2 1 equal to 0, now this we have define to be D is A 2 2 minus A 2 1 A 1 1 inverse A 1 2, so we have a compact expression A 2 1, A 1 1 inverse plus B 2 1 equal to 0.

(Refer Slide Time: 33:30)

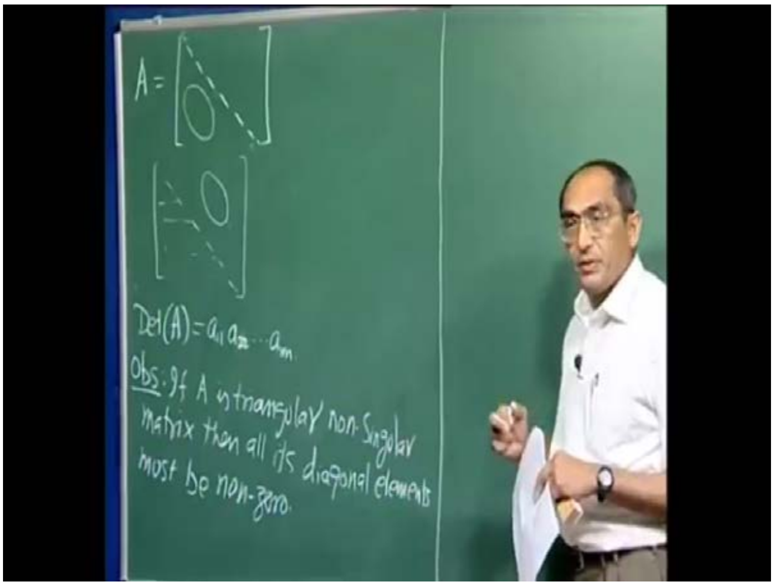


Here, inverse exists assumption we can get B 2 1 as minus D inverse A 2 1, A 1 1 inverse, so let us put the whole thing together and we get A inverse matrix as maybe I can put down as a as a lemma. If A 1 1 inverse and D inverse, which is same as A 2 2 minus A 2 1, A 1 1 inverse A 1 2 inverse exist, then inverse of A is if you recall was B 1 1, B 1 2, B 2 1, B 2 2 becomes the following B 1 1 is A 1 1 inverse 1 minus A 1 2. Now, need to plugging in B 2 in nut which we know is this, so that minus sign will become a plus sign D inverse A 2 1 A 1 1 inverse E 1, 2, 2, 1 inverse B 2 1, B 1 2.

We have the one D 1 2 is this which is minus A 1 1 inverse A 1 2 D 2 2, but B 2 2 is D inverse then D 2 1 B 2 1 is minus D inverse A 2 1 A 1 1 inverse, this is D inverse. So, we get a reduced form a recursive form were what we are doing is we are getting inverse n cross n matrix in terms of inverse of smaller metrics.

This expression may not be useful in case any one of these fail to have inverse if A 1 1 or D do not have inverse such a reduction is not much help. Although this expression is not going to lead to inverse for general case, but I am going to discuss a special case, where we can compute the inverse and that is for a triangular matrix.

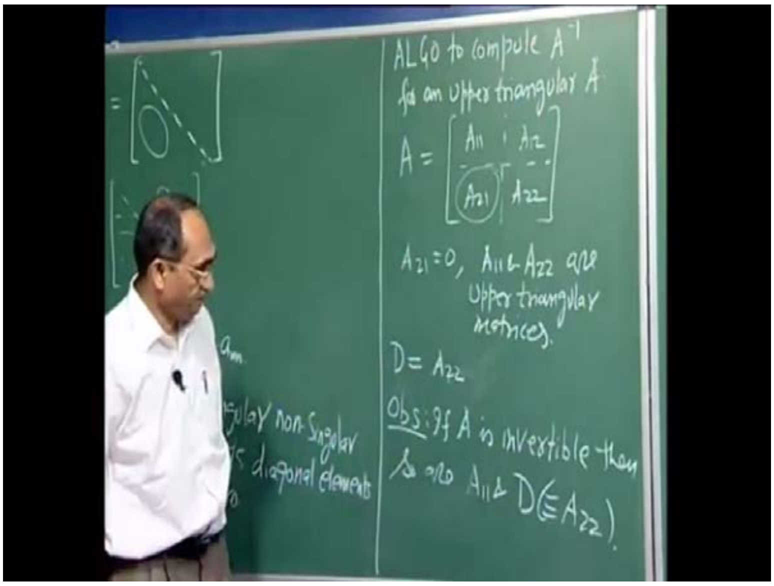
(Refer Slide Time: 37:23)



Suppose, A is a triangular, but triangular means is that if you happen to have a matrix in which all the elements of the diagonal an above may be non zero that only the elements in below the diagonals at 0. So, this may be non zero and these may be non zero, but these are all 0, then we call this an upper triangular matrix. Similarly, if the matrix as now one interesting thing about triangular matrix is that their determinant is very easy to the only contribution among n factorial towns in the determinant, which will definitely all others are going to be 0. They will contain this one of those 0s from hear the only term which will not contain any of the 0s will be the product of diagonal term.

So, the determinant of a where a is a triangular matrix upper or lower is nothing but A 1 1 times A 1, A 2 2 all the way to a n n, now if I am interested in computing the inverse of such a matrix I want the determinant to be non zero. Hence, it implies that all these diagonal element should be non zero, so one observation we will do make is that if A is triangular a non singular matrix, then all its diagonal elements must be non zero, now I am going to assume my A's and upper triangular matrix.

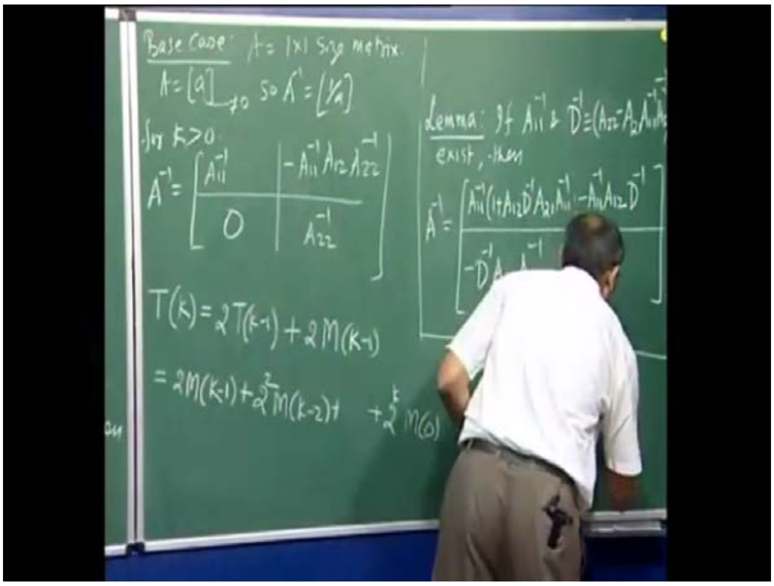
(Refer Slide Time: 40:34)



So, an algorithm to compute this inverse, so an algorithm to compute a inverse for an upper triangular A, now I am going to use over expression that we have that used and for that I need to split my matrix into half size. So, our A is 1 1, A 1 2, A 2 1, A 2 2, so what we immediately notice is that these entries are all 0, so A 2 1 is 0, the other two observations are this itself is an upper triangular matrix A 1 1 and shows this A 2 2 are upper triangular matrixes. Of course, by n 2 by n by 2 size, now let us try to write down the value of D this is because one of the matrices.

We are going to inverse D is A 2 2 minus 1 A 2 1 and so on which is 0, so this just reduces to A 2 2. Now, note is that to applied that result we need A 1 1 D invertible, now what do we notice here is that A 1 1 is this matrix. This is itself a upper triangular matrix, this is all the diagonal elements of the big matrix are non zero. So, all the diagonal element of this matrix are non zero and for this matrix hence A 1 1 must be invertible, because it is determinant is non zero, A 2 2 must be invertible. So, we make an observation here that if a is invertible then, so R A 1 1 and D which is same as A 2 2, you have just seen that now this immediately gives us or recursive algorithm.

(Refer Slide Time: 43:44)



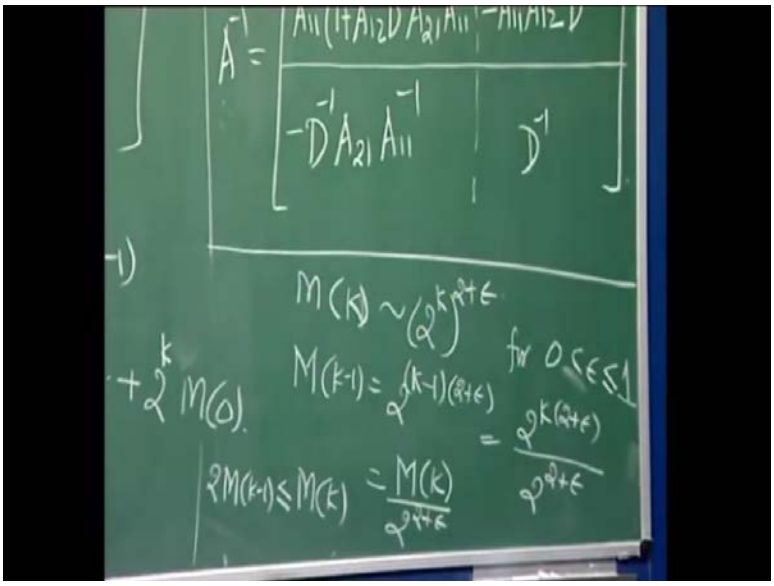
Now, base case is that our matrix A is a 1 cross 1 size matrix A 1 cross size matrix I is only one element that belongs to the diagonal. So, that can be non zero, now if this is invertible then that has to be non zero, so my matrix A looks like a and this is not equal to 0. So, we can immediately produce the inverse which is one over a notice that the plot it is nothing but a times 1 over x. So, base case is it takes constant time to compute the inverse n for general case a greater than 0 A inverse matrix looks like A 1 1 inverse 1 plus A 1 2 our D is A 2 2 inverse A 2 1 is 0.

So, that goes away and this term just vanishes just A 1 1 inverse this expression is minus A 1 1 inverse A 1 2 A 2 2 inverse because D is 2 2, this is A 2 2 inverse and notice here again we have A 2 1. So, this vanishes and we get this expression now this involves computation of two inverse of half size, but triangular matrix. So, one can recursive apply the same, so that is the end of the discussion, now let us take look at the cost of computation to compute the inverse of a triangular matrix of size 2 power k by 2 power k, I am supposed to compute two inverse of half size.

So, 2 t k minus 1, 1 of this and the other A 2 2 inverse, in addition we have to perform two multiplications of half size which I am going to denote by 2 m k minus 1, please note m k minus 1 denotes the cost of multiplying two 2 power minus k minus 1 matrices. We will say multiply these two and I will take the resulting the multiplying that with this matrix two of them this is different relation. We have to resolve and not going to worry about is exact value of this this could be done by using stresses algorithm, but there are better algorithm as well, so will just leave this as a black box and express this as m k minus 1.

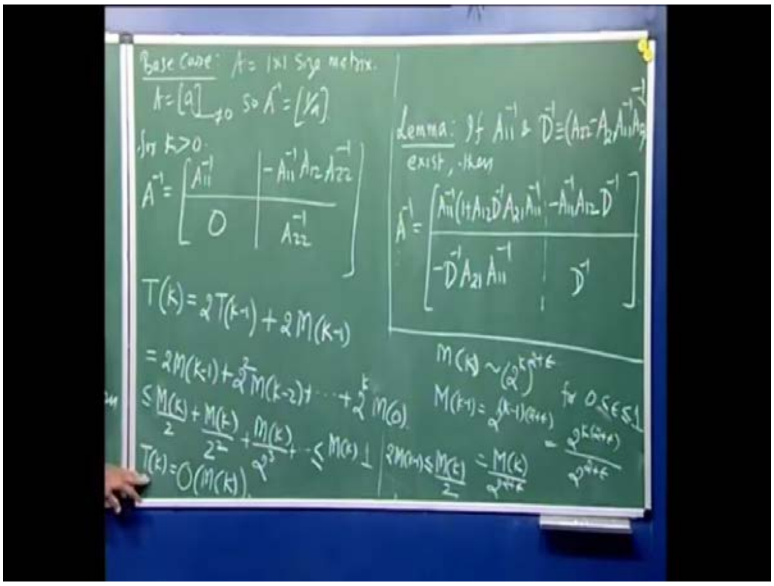
Now, let us solve this this turn out to be 2 m k minus 1 plus 2 square and k minus 2, 2 power k minus 1 m, it will be probably k m 0, this is the expansion of reference relation. Now, although I am not commenting on value of this one thing I do know is that this will take utmost 2 to the power k minus 1 cube that is our simplest algorithm. We have already seen a better one, but this can never take less than 2 to the power k minus 1 square because there are k minus 1 square entries in this matrix, so even just a somebody gives those entries just to output them will take that much time.

(Refer Slide Time: 49:03)



So, we will say that m k will be somewhere around 2 to the power k 2 plus epsilon, I can assume that for some epsilon m k is 2 to the power k power 2 plus epsilon. Then, epsilon is for somewhere between it cannot be 0, we know it is not also going to be one, but somewhere between. So, m k minus 1 is 2 power k minus 1 into 2 plus epsilon, which is 2 power k 2 plus epsilon over 2 power 2 plus epsilon so that I can sees in k divided by 2 power 2 plus epsilon. This is valid if we assume that this is equal to this, hence we can further say that 2 m k minus 1 is less than equal to m k. The reason is if I take two out and multiplying here, I still have 1 plus epsilon here, hence m k is greater than that, so we can use this in equal between m k and m k minus 1.

(Refer Slide Time: 51:08)



If I use that then this can be written as 2, it can be written as m k, notice that 2 m k minus less than equal to m k, this would be again m. So, I should say this is this is still because this will be 2 power 1 plus epsilon, so I will keep to like that way, I will like that as m k by 2 and this is m k by 2. This is m k by 2 square when I observed these 2 square, I will get m k by 2 square m k by 2 cube and that is less than equal to m k the sum of this geometric series is less than equal to one half by one minus one-half. So, it is no more than one, so this times 1, so what we conclude is that t of k is of the order of m of k.

It does not cost you more than just the cost of multiplying two matrix is of the same size, but of course, remember this is only for inverting a triangular matrix. Now, how to compute the inverse of a general matrix for that, in the next lecture we will discuss first how to break up a matrix into triangular matrices so that you can then compute the inverse.