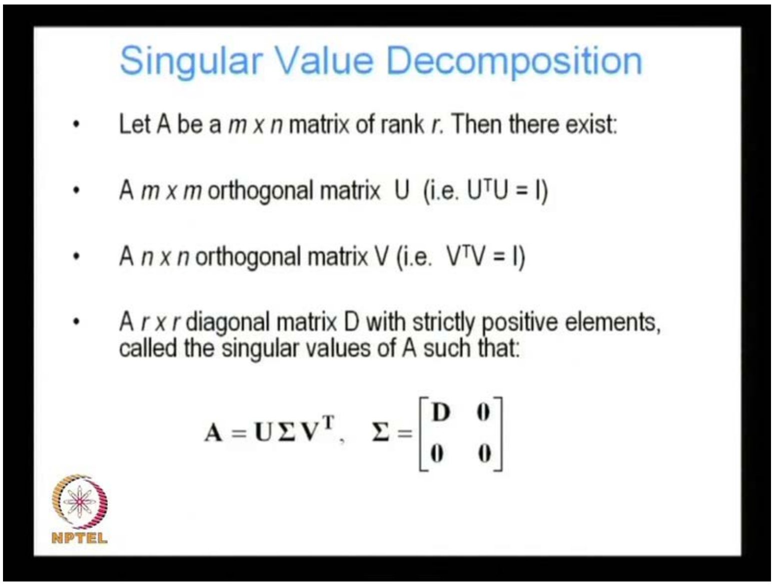
Numeric Methods in Civil Engineering
Prof. Arghya Deb
Department of Civil Engineering
Indian Institute of Technology, Kharagpur

Lecture - 4
Linear Systems – II

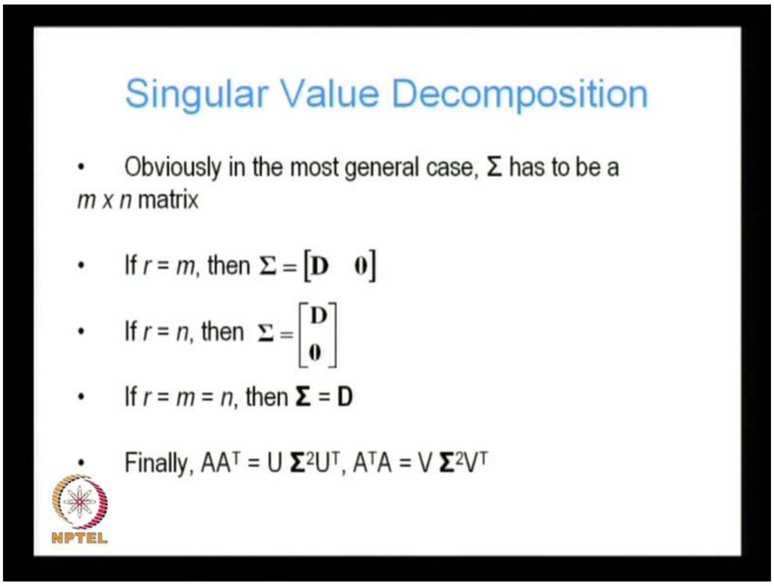
Going to continue our discussion on linear systems, this is the second part of our lecture on linear systems. Last time we talked about singular value decomposition, when we talked about how to find the singular values of an m by n matrix A, which is of rank r.

(Refer Slide Time: 00:33)



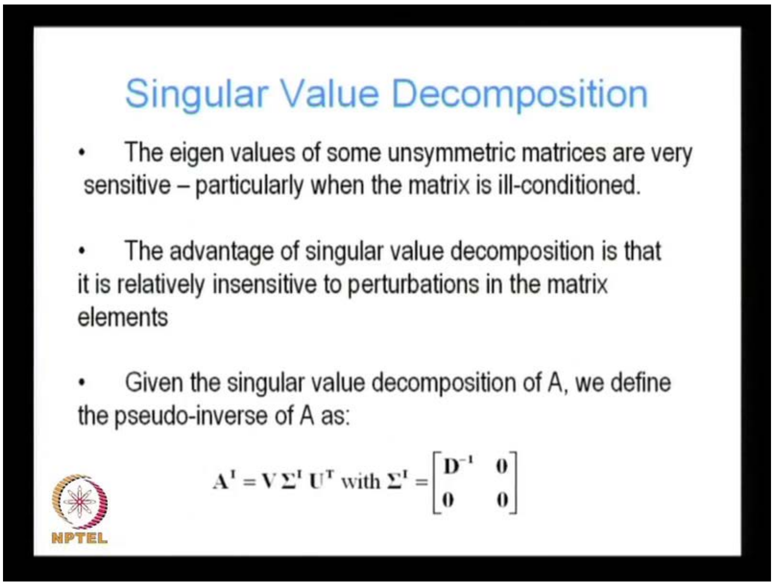
If A is an m by n matrix of rank r .We can be assured that there exist a m by m orthogonal matrix U, orthogonal matrix naming that matrix, transpose of that matrix in product with itself, that is U transpose U is going to be the identity matrix. An n by n orthogonal matrix V, that is, which is also an orthogonal matrix, which means V transpose V is again equal to the identity matrix I and an r by r diagonal matrix D, which has got strictly positive elements and those positive elements are called the singular values of A, so all these matrices U, V and D, when D is working as an expanded matrix sigma, they can be combined to form the matrix A. So, A is equal to U product, sigma product V transpose. So, U product sigma transpose V transpose is the singular value decomposition of the matrix A.

(Refer Slide Time: 01:53)



In the most general case, sigma has to be a m by n matrix; however, if the rank of the matrix is equal to m, that is the rank of the matrix is equal to the number of rows, in the matrix, then sigma is equals to D 0 right, however if the rank of the matrix is equal to n. The n is the column, number of columns in my original matrix, and then sigma is equal to D 0, where D is now in the form of a column vector. However, if sigma has rank r, which is equal to m, which is equal to n, which means that, A is the n by m matrix is a square matrix and it has got full rank then sigma is just equal to D. That is the matrix is no longer singular. It has got all its eigen values are positive real numbers. In that case, we can write. So, given the singular value decomposition of A, we can write AA transpose, as U sigma square U transpose similarly, by taking the transpose of that, we can write A transpose A is equal to V sigma square V transpose.

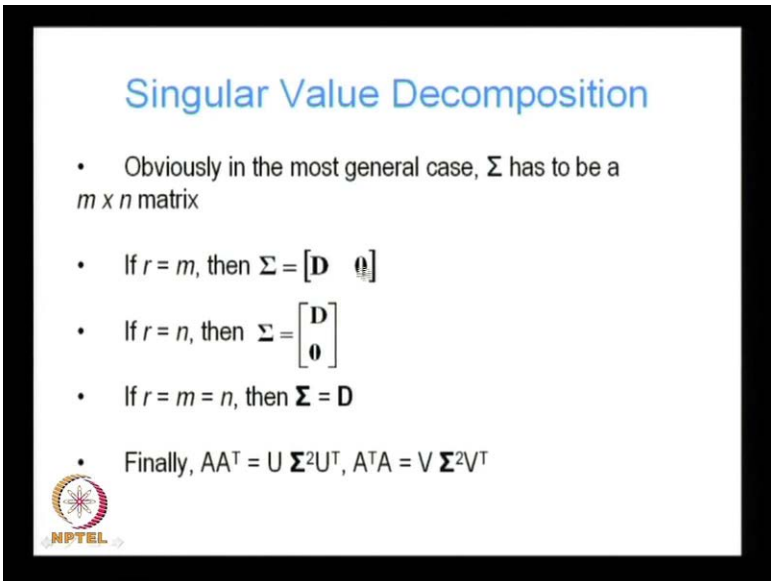
(Refer Slide Time: 03:20)



Why do we need singular value decomposition? The reason is, because many matrices particularly unsymmetric matrices their eigen values are extremely sensitive to small numerical perturbations. So, small numerical perturbations can lead to large change in the eigen values, such matrices a typically known as ill conditioned matrices, for such matrices singular value decomposition is an efficient way to find eigen values of that matrix.

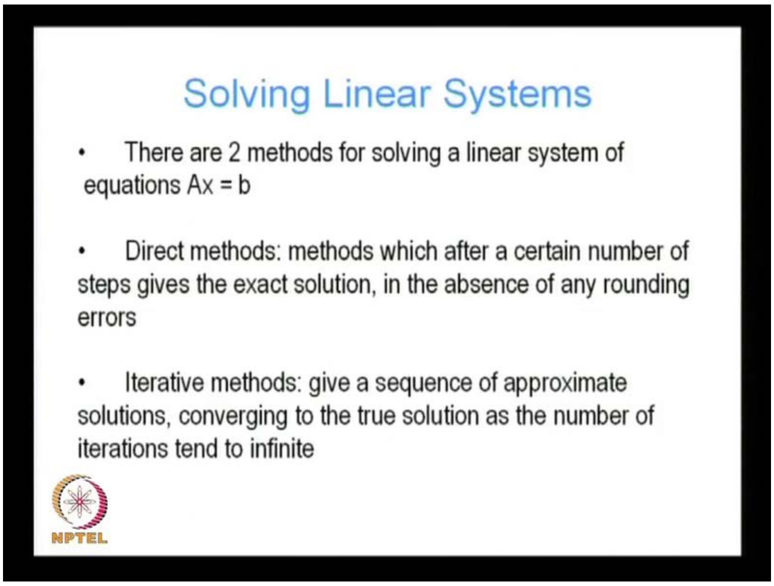
Given the singular value decomposition of A, we defined the pseudo-inverse of A, AI note that, this is different from A minus 1, where A minus 1 is the inverse. So, AI is the pseudo-inverse, the pseudo-inverse is again defined in terms of the orthogonal matrices V and U, as well as the diagonal matrix sigma. So, it can be written as the pseudo-inverse can written as V sigma I, U transpose with sigma I is the pseudo-inverse of sigma, which is equal to D inverse, 0, 0, 0.

(Refer Slide Time: 04:37)



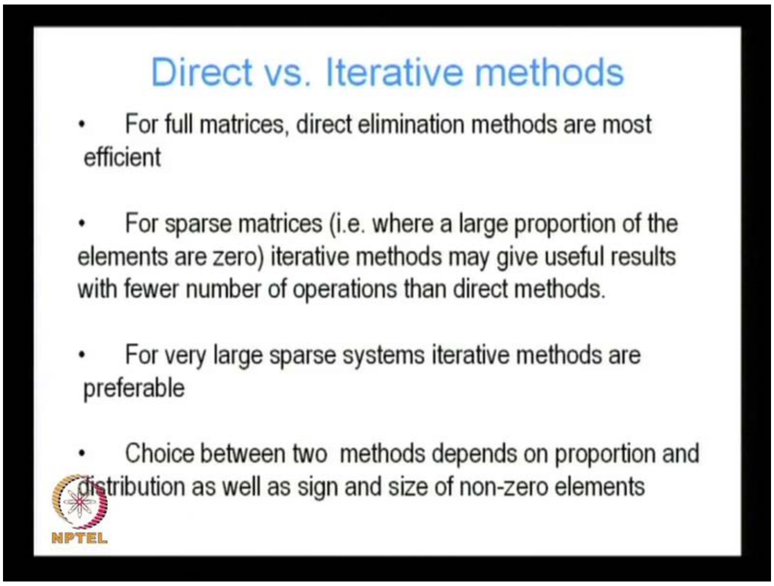
Let us recall, what was the form of sigma? So, sigma in its most general case had this form. So, the pseudo-inverse of sigma is D inverse, 0, 0, 0 in reality sigma does not have an inverse right, because this matrix has got 0 elements on its diagonal, which is singular matrix. It cannot be inverted, that is why we denote sigma I and call it pseudo-inverse rather than inverse. So, AI is the pseudo-inverse of A and we can write it as V sigma I U transpose with sigma I denoted as follows.

(Refer Slide Time: 05:30)



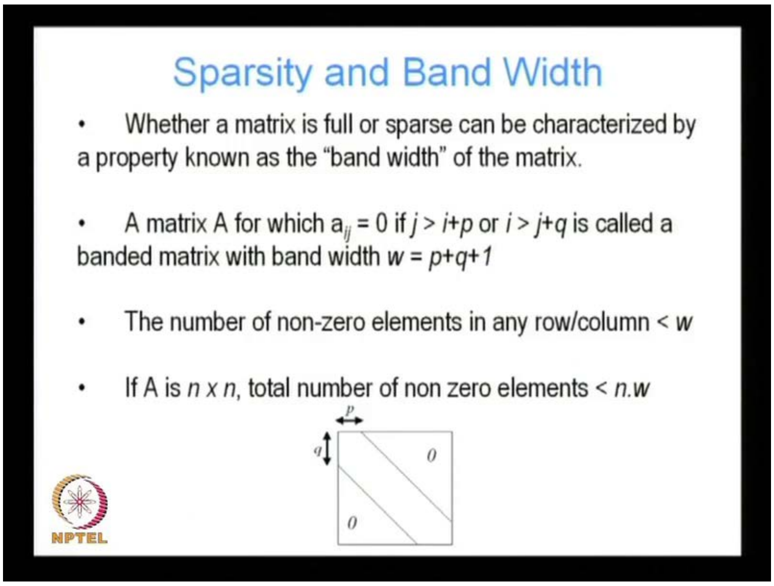
So, that was our discussion on solving as on singular value decomposition. As we discussed, it is very useful for solving problems, which are ill conditioned coefficient matrices. Let us, now go back to a discussion on linear systems and talk about certain general procedures for solving linear systems. They are basically two main methods for solving linear systems, one method is known as the direct method, the other method is known as the iterative method. In direct methods, we are assured that after a certain number of steps we are going to get the exact solution, in the absence of any round out and in case of iterative methods, we get a sequence of approximate solutions and if we continue the sequence to infinite number of if we continue the sequence in infinite number of times, we are going to get the exact solution or the true solution.

(Refer Slide Time: 06:37)



For, full matrices direct elimination methods are the most efficient; however, for sparse matrices, that is where a large proportion of the elements are 0, iterative methods make if useful results with the fewer number of operations than direct methods. For, very large sparse systems iterative methods are preferable. Therefore, what we get from this discussion is that, the decision on when to go for direct methods or when to choose iterative methods, typically depends on this sparsity of the matrix that is the number of non-zero elements in the matrix. So, depends on the proportion, distribution, as well as the sign and the size of non-zero elements.

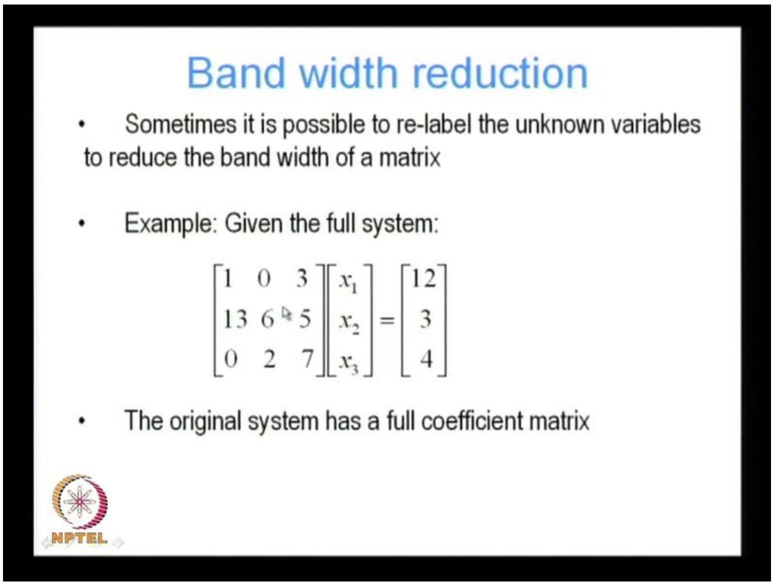
(Refer Slide Time: 07:31)



Whether, a matrix is full or sparse can be characterized by a property known as the band width of a matrix, a matrix A for which a i j is equal to 0, if j is greater than i plus p, it means that, all if we consider a particular row, then all elements in that row, which are more than p elements away from the principle diagonal are 0.

Similarly, if i is greater than j plus q, which basically means that, all elements in a column, which are more than q elements away from the principle diagonal are 0. So, if A is a finite strip bounding the principle diagonal in the neighbourhood of the principle diagonal. The matrix as non-zero elements, while everywhere else; the elements of the matrix are 0 and in that case, we say that the band width of the matrix is W, the W is equal to p plus q plus 1. So, the band width of the matrix is p plus q plus 1 denoting the principle diagonal. So, this is the band width of a matrix. The number of this automatically gives us the total number of non-zero elements in the matrix, which has to be less than n dotted with W.

(Refer Slide Time: 09:11)



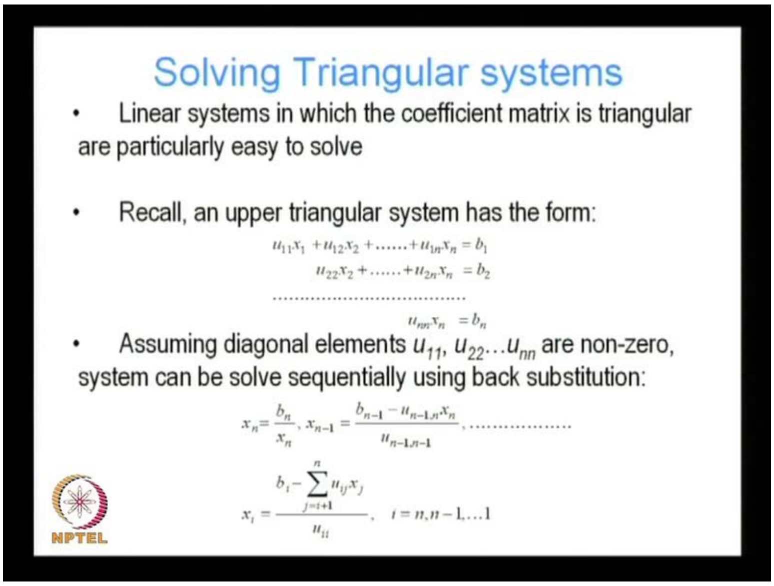
Sometimes, it is possible to re-label the unknown variables in a matrix, to reduce the band width of a matrix. Consider this 3 by 3 system it is clear, where this matrix has full band width that is, it is band width is 3 for instance in the first row, in first row the third column is non-zero. So, it has got full band width. So, the band width of this matrix is 3. So, it has got a full coefficient matrix; however, we will see that by re-labelling the variables it is possible to reduce the band width of this matrix.

How is that possible? Well, if we re-label X2 as X3 and X3 as X2, we get a band limited system with a band width of 2 instead of 3. Now, you can see that the maximum number of non-zero elements in a row is two elements away from the principle diagonal, this is the last element are 0, now similarly for the row for the column. So, this is my principle diagonal, we have one non-zero element away from the principle diagonal, beyond that we have 0 elements.

So, now the band width of the matrix is two rather than three, algorithms to reduce band width. It can be used for large matrices, this is because solution of band limited matrices is much less expensive than full matrices for example, there is the Cuthill-Mckee algorithm, which is a widely used algorithm for the numbering, the equations in order to reduce the band width.

Similarly, there are more advanced versions of the Cuthill-Mckee algorithm there are several other algorithms, which can be used for band width minimization. One thing we note is that even, if an original matrix is band limited it is inverse may be full therefore, if we work on the original matrix and we use a band width minimiser and the original matrix and reduce its band width, but then we try to inverted we might end up with the full matrix. So, all are work on band width minimization would be of no use, because eventually we have to we have to use that invert inverted matrix to solve our system of equations and if the inverted matrix is full that is going to be as expensive. It is not going to reduce the computational expense at all.

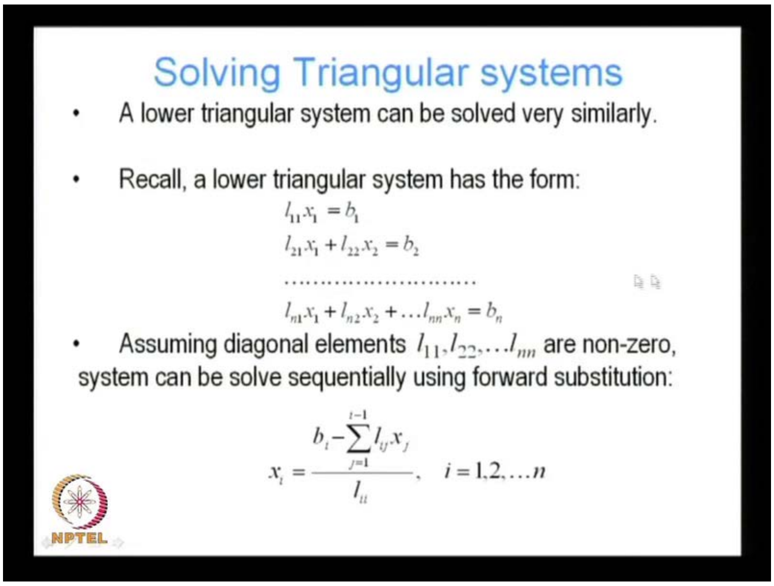
(Refer Slide Time: 12:05)



Next, we would like to look at particular types of matrix, which are very useful with for solving linear equations, which give us a form, which is very useful for solving linear equations and these are known as triangular systems. Recall an upper triangular system has the following form: U11 x1 plus U12 x2 through U1n xn is equal to b1, which is the first equation in my system. The second equation has U22 x2 plus U23 x3 through U2n xn is equal to b2 and so on. So, forth until the last equation is just Unn xn is equal to bn. So, it is clear that in upper triangular system have all elements below the principle diagonal 0, an all elements at or about the principle diagonal non-zero. So, this system can be solved very simply for instance. We can easily invert the last equation, we can solve for xn as bn by unn. So, it just one division which gives xn.

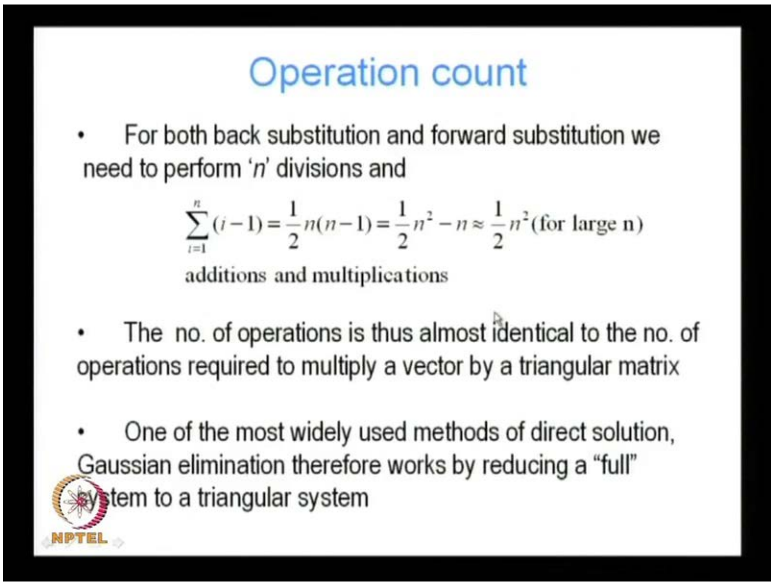
So, once I know xn I can use the n minus 1th equation to solve for xn minus 1, because xn minus 1th equation has the form, un minus 1, n minus 1, xn minus 1 plus un minus 1 n xn is equal to bn minus 1. So, if we do xn we can solve for xn minus 1 using this simple equation just by one division, one addition, and one multiplication, and one addition. Similarly, we can start from the end we go progressively backwards and we can use this general formula xi is equal to bi minus sigma uij xj, j is sum do where i plus 1 to n divided by uii for i is equal to n through one to solve this entire system. So, we start with the simplest equation and worked have a backwards to solve for the remaining variables.

(Refer Slide Time: 14:31)



Similarly, if we have a lower triangular system, that can also be solved very easily. The lower triangular system is one, which all elements at or below the principle diagonal is non-zero that all elements above the principle diagonal are 0. In this case, the last case was backwards substitution, because we solve the last equation and worked have a backwards. While in this case, we solve the first equation, which is easy enough to solve x1 is equal to b1 by l 11 and work have a forward. We solve for x1, we solve for x2, we solve for x3 and so on. So, forth, but important thing to note is that, this assumes that the diagonal elements l 11, l 22 through l nn are non-zero, only when these diagonal elements are non-zero, can the system be solved sequentially using either forward substitution or backward substitution.

(Refer Slide Time: 15:45)



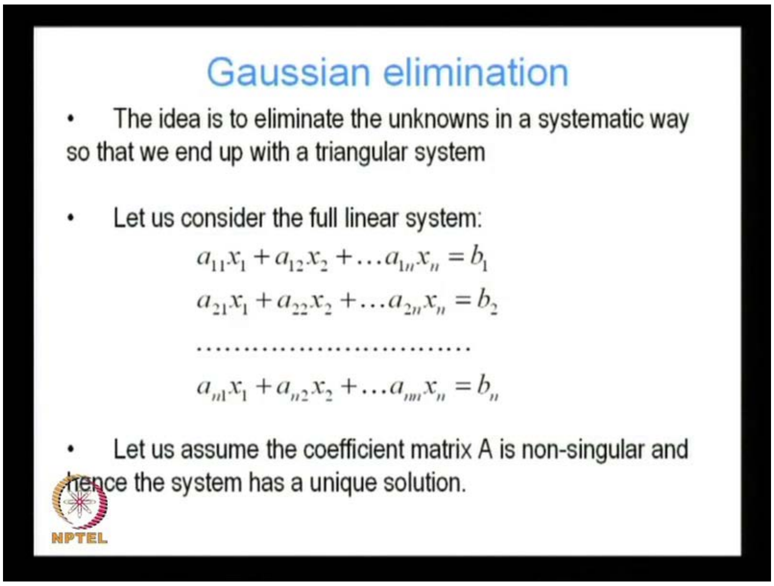
For, both back substitution and forward substitution, we need to perform n divisions. Why do we need to perform n divisions? So, that for each equation, we have to perform one division. So, n divisions and we have to perform for each equation sigma i is i minus 1 additions or multiplications basically let us take, a look at our equations and I think that would make it clear. For instance, for solving the first equation, we just have to perform one division. For solving the second equation, we have to perform one multiplication and one subtraction and one division. Similarly, for solving the third equation, we have to perform one division, but then we have to perform two additions and subtractions followed by two multiplications and two additions and subtractions.

So, the total number of additions and multiplications, for the first equation, we have for i is equal to 1, we do not have any multiplications additions, so 0 additions and multiplications. For the second equation, we have i is equal to 2, so one addition and one multiplication. For the third equation, we have i is equal to 3, so two additions and two multiplications and so on. If, we keep adding the number of additions and multiplications, we will find the total number of additions and multiplications is equal to 1 by 2 n square minus n.

And if a large systems, n square is going to be much larger than n. So, 1 by 2 n square minus n can be written as approximately equal to 1 by 2 n square. Therefore, backward substitution and forward substitution, we need to perform 1 by 2 n square additions and multiplications. But since, and the number of divisions is n, but again n is comparatively much smaller than 1 by 2 n square. So, the total operation cost approximately is 1 by 2 n square. The number of operations 1 by 2 n square is thus almost identical to the number of operations required to multiply a vector by a triangular matrix. Why that is the triangular matrix has only 1 by 2 n square elements right. So, if I multiply a triangular matrix with the vector. I have to perform 1 by 2 n square operations. So, the number of operations is identical to the number of operations required to multiply a vector by a triangular matrix.

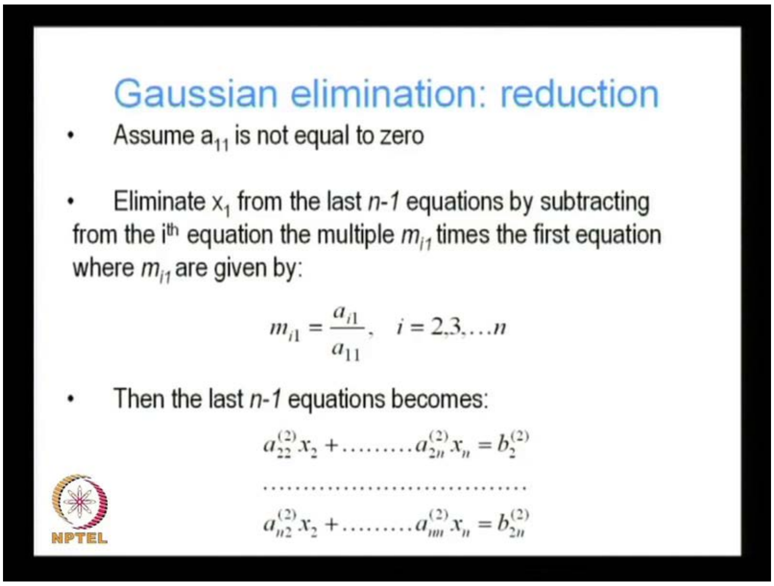
One of why do we talk, so much about triangular systems, because to solve an equations using Gaussian elimination, triangular systems are probably the most efficient way of solving it. So, the basic idea is that we try to reduce the system to a triangular system. So, the first part in any Gaussian elimination is attempts to reduce a full system to a triangular system and once it has reduced it to a triangular system it uses forward substitution or backward substitution to solve the system.

(Refer Slide Time: 19:27)



So, the idea is to eliminate the unknowns in a systematic way. So, that we end up with a triangular system. Let us consider the full linear system, which is an n by n system with the coefficient matrix is of size n by n, a 11 x1 plus a12 x2 through a 1n xn is equal to b1. Similarly, a 21 x1 plus a 22 x2 through a 2n xn equal to b2 and so on. So, forth let us also assume that, the coefficient matrix A is non singular and hence, the system has a unique solution. We recall our discussion from last lecture, when we talked about the conditions, which are necessary for a system of equations to be non singular.

(Refer Slide Time: 20:21)

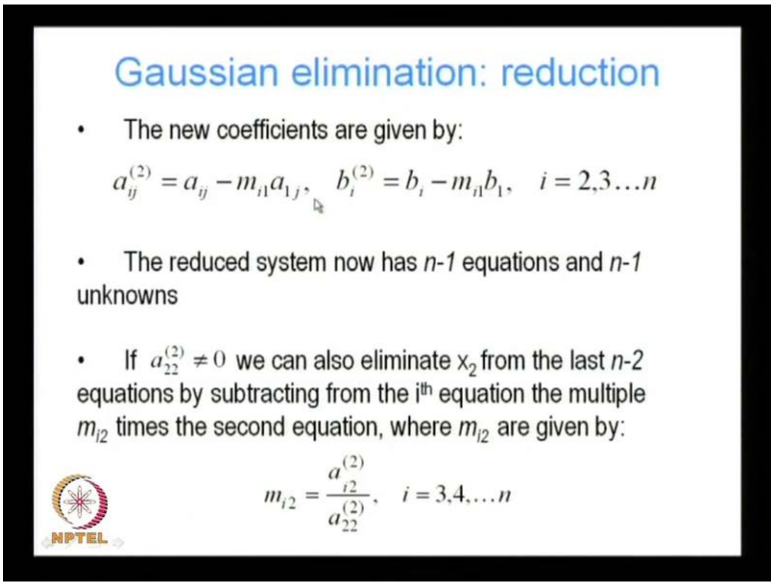


We also assume that a 11 is not equal to 0, if we eliminate x1 from the last n minus 1 equations by subtracting from each equation. The multiple m i1 times the first equation, where m i1 is given by a i1 by a 11 then, the last n minus 1 equations do not have any variable x1. So, the size of the system gets reduce to n minus 1 by n minus 1.

Let's go back and take a look at a full system and see what, we are talking about. So, what we are saying is that, the first equation, we divide all the elements by a 11 then, we add a 21 times the first equation. We subtract a 21 times the first equation, from the second equation. We subtract a 31 times the first equation, from the third equation. We subtract n a n1 times the first equation from the nth equation. If we do that the first coefficient always become 0, because we are dividing the first equation by a 11.

So, the coefficient of x1 in the first equation is 1. If, we multiply this by a 21 and subtracted from the second equation, the coefficient of x1 is going to vanish. Similarly, we do the same thing for the third, fourth, fifth and so on so forth until the nth equations. If, we do this then, we are going to get the last n minus 1 equation and no longer going to have a x1 term. So, the coefficient to the x1 in the last n minus 1 equation, is going to become is 0 and we are going to get that equation. We are going to get n minus one equation in terms of n minus 1 variable x2 through xn.

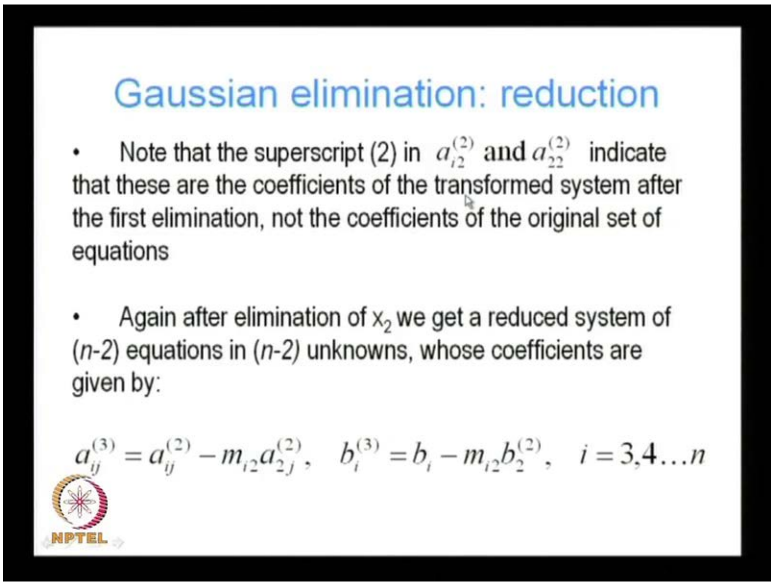
(Refer Slide Time: 22:28)



So, the new coefficients are given by this, it is just in short form in symbolically. So, the reduced system now has n minus 1 equation and n minus 1 unknown. If, at the second stage after the first reduction, we are left with the n minus 1 system and if it turns out that this a 22 is also non-zero then, we can continue with the same process throughout. So then, what we are going to do? We are going to divide this equation by a 22. So, the first coefficient is going to become 1 and then we are going to multiply this equation by the first coefficient of x2, in the second equation and subtracted from the second equation.

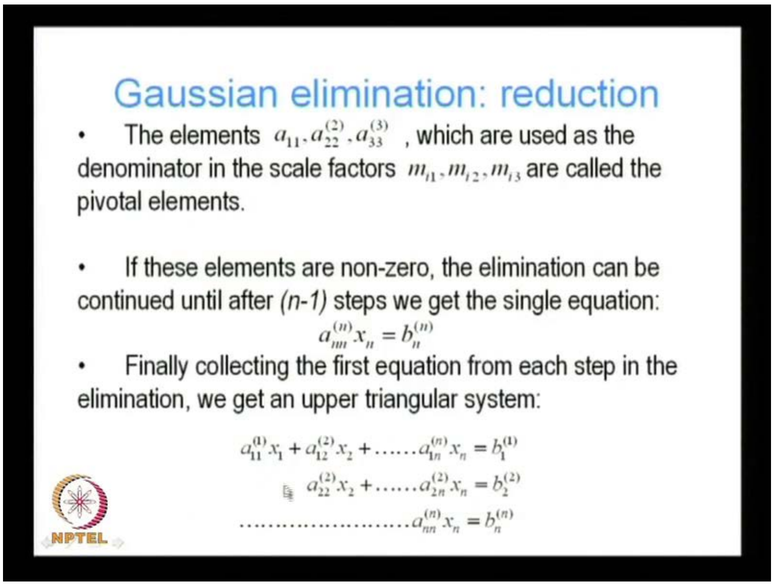
So, in that case, the second equation is going to have 0 as the coefficient of x2. Similarly, we will do the same thing for the third equation and we will continue for up till the nth equation. So, if we do that, if we can also eliminate x2 from the last n minus 2 equations by subtracting from the ith equation, the multiple m i2 times the second equation, where m i2 is given by a i 22 by a 222, where i is equal to 2, 3, 4 through n. The subscript 2 denotes the step in the reduction process. So, 1 denote the first step, 2 denotes the second step.

(Refer Slide Time: 24:22)



These are the subscript 2 in a i2 2 and a 22 2 indicates that these are the coefficients of the transformed system, after the first elimination. Similarly, we can continue this process, for the thirds for the third step and so on through the end steps. Until, we get a system like this, which we will call is an upper triangular system.

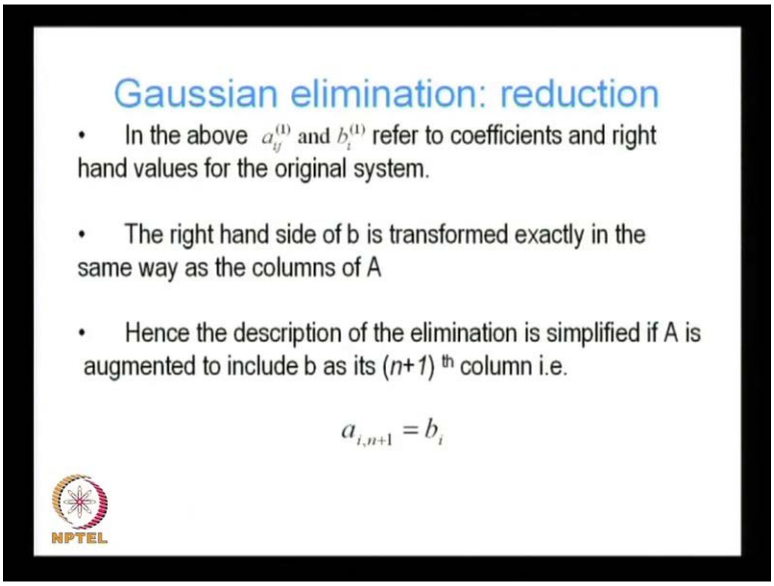
(Refer Slide Time: 24:45)



So, from the second equation, we have made sure that the coefficient of x1 is 0. From the third equation, we have made sure that the coefficient of x1 and x2 are 0. Similarly, until we have continued with that process until, we have reach the nth equation, where we have made sure that coefficients of x1 through xn minus 1 are 0 and the only variable with non-zero coefficients is xn.

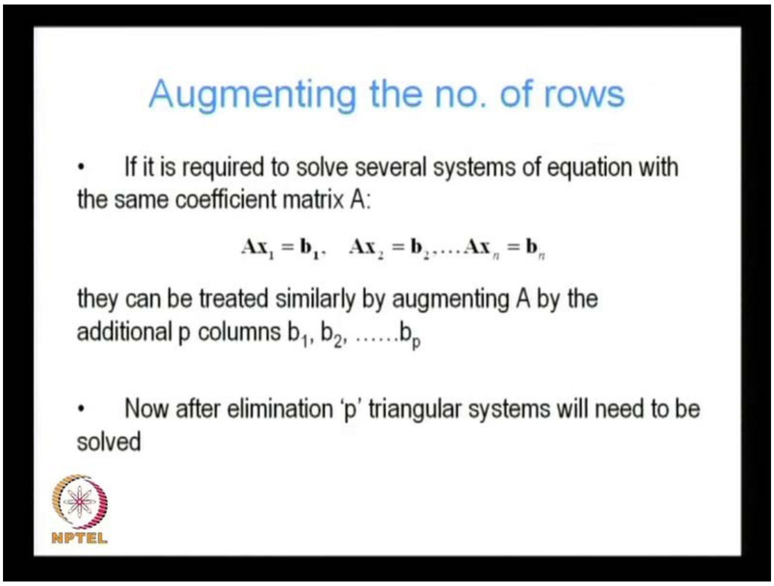
So, we have reduce this system to an upper triangular system, when we have got a single equation a nn n xn is equal to bn n again, we can solve this equation by back substitution. We solve for xn substitute that in the n minus 1th equation to get xn minus 1, substitute xn and xn minus 1 in n minus 2th equation to get xn minus 2 and so on. So, forth until, we get all the variables x1, x2, x3 to through xn.

(Refer Slide Time: 26:05)



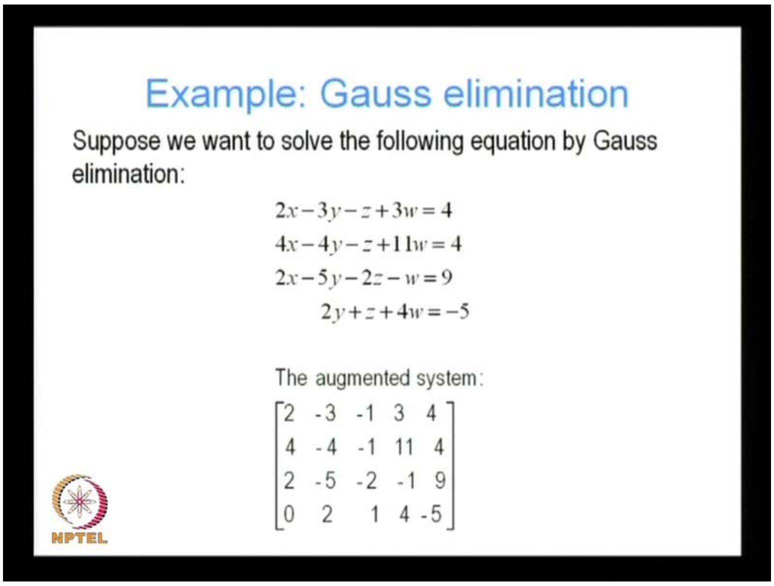
So, in the above a ij 1 and bi 1 refer to coefficients and right hand values for the original system, bi 1 refers to the right hand values, after the first step in the reduction procedure. Similarly, bi n would refer to the right hand values after the nth step in the reduction procedure. The right hand side of b is transformed exactly in the same way as the columns of A, hence the description of the elimination is simplified. If, A is augmented to include b as it is, n plus 1th column. Basically, we move the right hand side to the left hand side with of course, with the negative sign and that resulting system is called an augmented system.

(Refer Slide Time: 27:07)



So, if it is required to solve several systems of equations with the same coefficient matrix A, that is we are interested in solving several systems Ax1 equal to b1, Ax2 equal to b2 through Axn equal to bn, that is the right hand side is changing, but the coefficient matrix remains the same. Then we can augment A by including the additional p columns b1, b2 through bp and increasing the size of the matrix from n by n to n by n plus pn by n plus p.

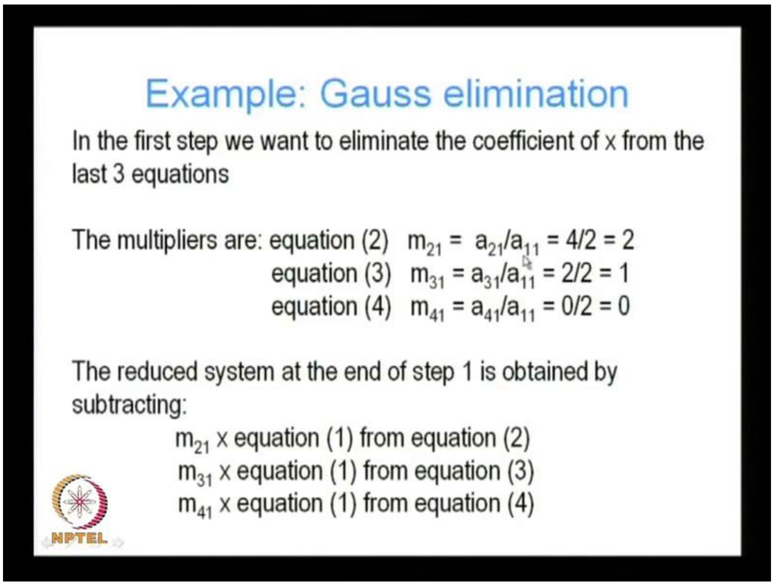
(Refer Slide Time: 27:57)



So, after elimination, we will have to solve p triangular systems for each of the right hand sides b1 through bp. Let us, look at a simple example, suppose we want to solve the system of equations with four equations and four unknowns and we want to solve it using gaussian elimination. The augmented system is given by this, where we have moved right hand side to the left and we haven't at and we are going to operate on this using gaussian elimination, with first step will be the reduction and then, we will use back substitution to solve this system.

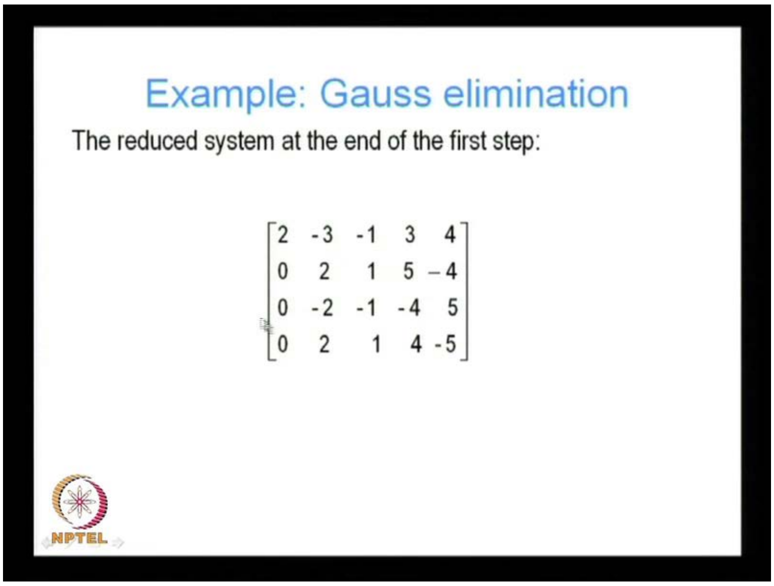
So, in the first step, we want to eliminate the coefficients of x in the second equation right. So, this 4x we want to get rid of this 4x. So, what do we do? We divide the first equation by 2, because 2 is the coefficient of x in the first equation, we multiply the after dividing the first equation by 2, and we multiply the first equation by four and subtract that from the second equation. Similarly, we multiply the first equation by 2 and subtract it from the third equation and so on.

(Refer Slide Time: 29:20)



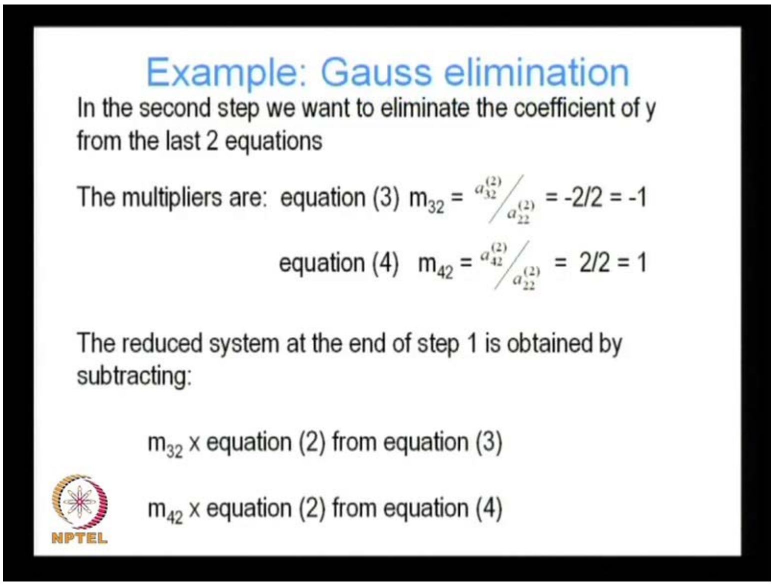
So, if we do that the multipliers for equation 2 is m 21 is equal to a 21 by a 11, which is 4 by 2 is equal to 2. The multiplier for equation 3 is a 31 by a 11. So, 2 by 2 that is equal to 1. And the multiplier for equation 4, where the coefficient of x is 0 is nothing but, 0 by 2 which is equal to 0. So, if we subtract m 21 times equation 1 from equation 2, m 31 times equation 1 from equation 3 and m 41 times equation 1 from equation 4.

(Refer Slide Time: 30:10)



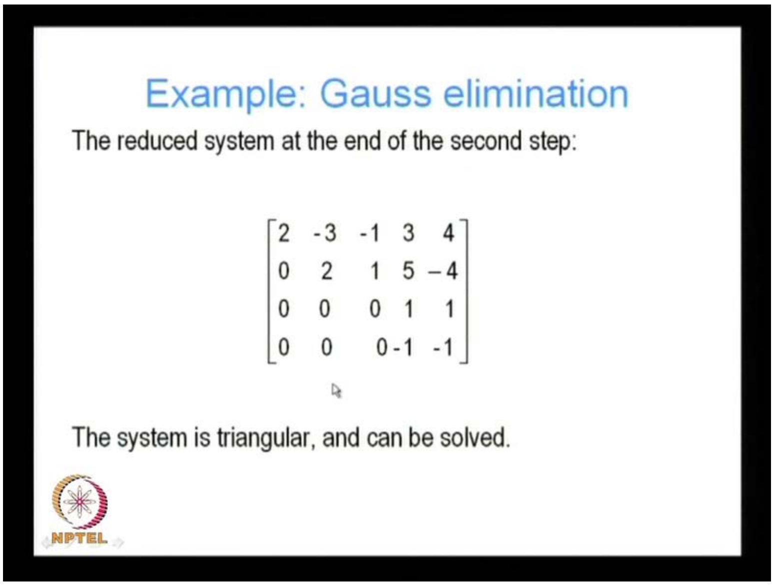
Then, we are going to get this system. This is the reduced system at the end of the first step then, you can see that the second equation. Now, have 0 coefficients it has got a 0 coefficient as on the first column. So, the coefficient of x1 is actually equal to 0.

(Refer Slide Time: 30:33)



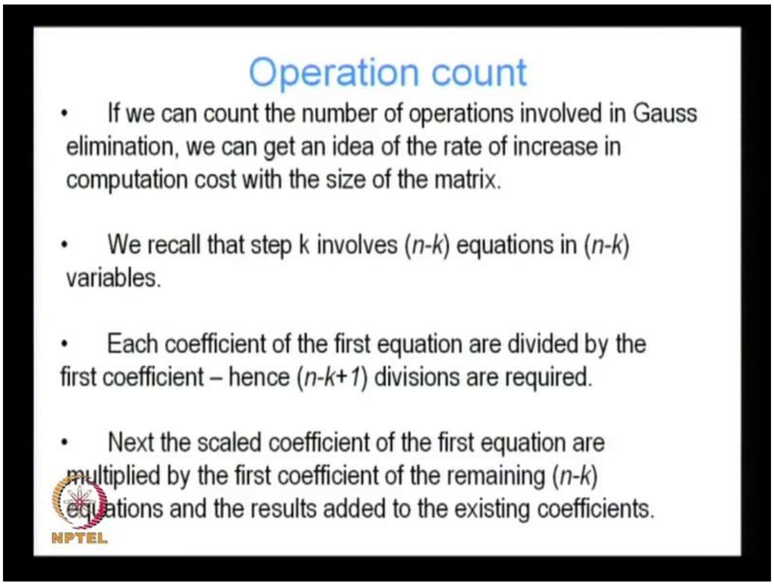
In the second step, we want to eliminate the coefficient of y from the last two equations. So, this is the coefficient of y. So, I want to eliminate this, from the last two equations. So, what do I do? I divide the second equation by 2 and then, I multiply the second equation by minus 2 and added to the third equation. I multiply the second equation by 2 and add it to the fourth equation and if I do that. So, basically my multipliers, now are a 32 2 by a 22 2. So, minus 2 by 2, this is equal to minus 1, for equation 3. For equation 4 my multipliers are m 42 is equal to a 42 2 by a 22 2, which is 2 by 2, which is equal to 1. And then, I obtain my reduce system by multiplying equation 2 by m 32 and subtracting it from equation 3 and multiply equation 2 by m 42 and subtracting it from equation 4, then after, I do that.

(Refer Slide Time: 31:55)



This is my system, this is my resultant system and as you can see, this system is triangular right. If you look at my coefficient matrix, it is a triangular system and this can be easily solved using back substitution. Actually for, if this matrix were different we might have had to do a third step, but the structure of the matrix is such that even after, two steps of reduction we get a fully triangular system and we can go ahead with the back substitution and can solve the system.

(Refer Slide Time: 32:42)

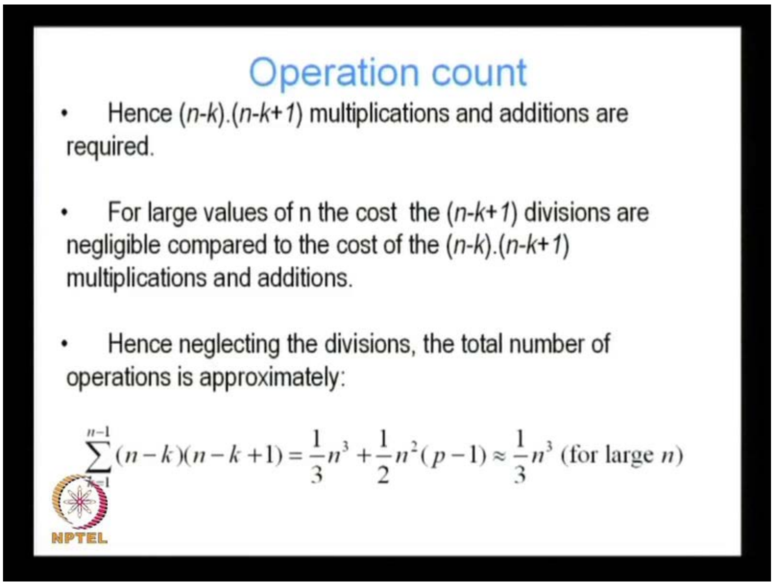


Let us do an operation count, basically we want to count the number of operations involved in gauss elimination. Why do we want to do this, we want to get an idea of the rate of increase in computational cost with the size of the matrix. So, as n increases how quickly does the computational? What is the weight of increase of the computational cost? That is, what we want to find out, to do that we have to count the number of operations in each step.

We recall that step a, k involves n minus k equations in n minus k variables for instance, when after the first step we have n minus one equations and we have n minus one variables. The second step has n minus two equations, n minus two variables and so on. So, forth each coefficient of the first equation are divided by the first coefficient hence n minus k plus 1 division are required, because there are n minus k terms and then, there is a right hand side also. So, the n minus k plus 1terms and each term in that first equation has to be divided by the first coefficient. So, the n minus k plus 1 division are required.

Next, the scaled coefficients of the first equation are multiplied by the first coefficient of the remaining n minus k equations, and the results are added to the existing coefficients.

(Refer Slide Time: 34:30)



So, what does that mean; that means, there are n minus k times n minus k plus 1multiplications and editions, why the first equation, we have already, we disregard the first equations. So, now, the n minus k equations, but for each n minus k equation we have to do n minus k plus 1 multiplications and additions. So, n minus k times n minus k plus 1 multiplication and additions are required.

For, large values of n the cost of the n minus k plus 1 divisions are negligible, compared to the cost of the n minus k times n minus k plus one multiplications and additions, why is that, because this involves n square terms. So, the cost of this is negligible compared to the cost of the n minus k plus 1 divisions hence, neglecting the divisions the total number of operations is n minus k times n minus k plus 1, but remember this has to be performed for each step in the reduction and there are n minus 1 steps. So, we sum over k is equal to 1 to n minus 1 and if we do that, we get one third n cube plus 1 by 2 n square p minus 1 p is a. You remember is the number of right hand sides the sake of simplicity, if we assume that the number of right hand sides is 1. So, we have 1 by 3 n cube approximately 1 by 3 n cube operations for large n.

Even if, p is not1 by 3 n cube would be much greater than 1 by 2 n square times p minus 1 again, because n cube is much greater than n square. So, the total number of operations is about 1 by 3 n cube for large values of n.

So, once we have obtained a triangular system, to solve the triangular system, it takes n divisions and as we saw earlier. It takes 1 by 2 n minus 1 times n addition or multiplications, which is approximately equal to 1 by 2 n square operations, 1 by 2 n square additions and multiplications hence, the back substitutions for large systems requires 1 by 2 n square additional operations.

Since, for large systems 1 by 3 n cube recalled, that the from our previous slide 1 by 3 n cube is the total cost of the reduction, 1 by 3 n cube is always going to be much larger than 1 by 2 n squared for large systems. So, the major cost in Gaussian elimination lies in the reduction to the triangular system. So, if you want to speed up the cost of gaussian elimination. We have to find out cheaper ways to do the reduction, we have to cut down the cost of the reduction and there are several ways of doing that, we are going to talk about some of them later on.

But recall our discussion of Gaussian elimination. We always impose the condition, that pivot the term by, which we divide those equations. The term by which, we normalize the equation for instance. Let us go back and look at our equations again. So, these pivots right a 22 2 or for instance in this case a 11, these pivots have to be non-zero. If they become 0 then, this procedure stops. We cannot proceed with a reduction procedure. So, the prerequisite for Gaussian elimination to work without any interchange of rows or columns is that these pivots must be non-zero. So, from the algorithm for Gaussian elimination, we have considered. So, far it is evident that if any of the terms a 11, a 22 2 or a 33 3 all of these terms are the pivots right. The terms by which we normalize the equation if these are 0 then, our algorithm is going to fail.

Hence, all practical applications of Gaussian elimination require some pivoting, because we cannot be assured, that these terms are not going to be 0 to an understand how the pivoting works. Let us consider the system, the simple system 3 by 3 system in 3 unknowns, x1 plus x2 plus x3 is equal to 1, x1 plus x2 plus 2 x 3 is equal to 2 and x1 plus 2x 2 plus 2x 3 is equal to 1, after the first step of the elimination, we get 0 x2 plus 1x 3 is equal to 1 and 1x 2 plus x3 is equal to 0.

Since, the coefficient of x2 in the first equation is 0, this algorithm is going to break down, because the pivot is 0; however, the problem is resolved, if we interchange the rows in the above equation. So, if we interchange the rows in the above equation since, the coefficient of x2 in the second equation is positive. We can again proceed with the reduction.

So, in general suppose at step k in our reduction process we find a kk k is equal to 0. In that case, there will always be some other element in the kth column suppose a rk k, which is non-zero go back and look at our previous example, in this case a 22 k is 0, but a 32 k is no longer is not 0. So, that is why we can interchange these two rows to make sure, that the pivot is no longer 0.

Why are we assured that, there must be at least one element in the kth column, which is not 0, because if all the elements in the kth column is 0, this implies that the first k columns are linearly dependent, because we have form the kth column by to reach the kth step, we have form the kth column by taking scalar multiples of the first k minus 1 columns, so if the kth column, if all the terms in the kth column as 0. It only means that I can first k minus 1 columns are linearly dependent, recall our discussion of linear dependence and liner independence from our previous lecture.

So, if k minus 1 vectors, if we can add k minus 1 columns that is, k minus 1 vectors. If, we multiply each column by scalar and we add them together and the result is 0. We say that those first k minus 1 columns are linearly dependent, they are not linearly independent and if their linearly dependent, what does that mean; that means, that coefficient matrix A is singular, because it has it is rank is less than n.

But since, we have assumed that the matrix is non-singular; that means, it has got full rank that is the kth column cannot be obtained as a linear combination of the first k minus 1 columns. So, we are assured that, at least some elements in the k minus 1 column must be non-zero otherwise, the matrix is singular. Since at least, one of the elements in the kth column is non-zero, we can always interchange that row with the kth row to restore not to make sure, that a pivot is non zero and we can continue with reduction proceeding.

Since, a rk k is non-zero. We can interchange rows r and k and proceed with elimination thus it follows that any non-singular system of equations can be reduced due to triangular form by gaussian elimination, accompanied by row interchanges. If the system is singular, there is no assurance this can happen because we can get eventually a row, which has got 0 entries all 0 entries and in that case we can no longer get a non-zero pivot.

However, it is not always true that only, if the pivot term is 0, it is necessary to do pivoting. If the pivot terms, becomes really small it becomes sufficiently close to 0 that if we divide the equation by that pivot term, we are going to get numerical instability. In that case, also it might be necessary to do pivoting. So, it may be necessary to perform row interchanges not only when the pivotal element is exactly 0, but also when it is nearly 0 in order to improve numerical instability to see the need for this we consider the previous system the previous system which we considered earlier, but in this case we change the coefficient of a 22 from one to 1.0001.

Let is take a look at our previous system. So, here the coefficient of a 22 was 1. So, instead of change if keeping it 1, let us change it to 1.0001 and see what happens, if we do that, we will see that our triangular system after Gaussian elimination is going to become something like this, and if we do that substitution and using round off after three decimal. So, we will get x1 equal to 0, x2 is equal to 0, x3 is equal to 1; however, the true solution if we use four decimals, if we round to four decimals is actually x1 equal to 1, x2 equal to minus 1.0001, x3 is equal to 1.0001.

So, we can see by using a nearly 0 element as the pivot our system of equations has got a totally erroneous solution, why because when we use the pivot 1.0001. Our system became ill conditioned. So, minor perturbations give totally different solutions. So, if instead of using as near 0 pivot if we did pivoting if we interchange the rows to get sufficiently large pivot, we would have got the true solution.

So, it is very important to do pivoting not only when the pivotal element is fully 0, but when the pivotal element is nearly 0. It is clear that without interchanges Gaussian elimination is going to be inherently unstable and will give rise to meaningless results;

however, by interchanging rows 2 and 3 in our in rows 2 and 3 in this equation we could and again using three decimal points in our computations, we could get x1 is equal to 1 and x2 is equal to minus 1.0001, which is almost exactly the true solution thus it is clear that interchange of rows is essential in order to obtain accurate solutions in such situations.

Now, let us talk about two particular types of pivoting first, we want to talk about partial pivoting. If, we interchange of rows is extended to make sure that, we interchange the kth row with the row that has the largest element in the kth column, it is known as partial pivoting, this is clear if we look at this little picture. Suppose, we are looking at this system we have reduced it up to here. So, it is a triangular system up to here and this is our and we have here left with this system and we want to reduce it further and we look at the pivotal element here and we find that the pivotal element is either 0 or sufficiently close to 0, so that we need to do pivoting.

So, what do we do? We look at all the elements in the shaded region that is basically we look at all the elements in this column and we select the element which has got the largest non-zero magnitude, which has got the largest absolute value and then we interchange this row with the top row. So, that our pivot we get the largest with the best estimate we can of the pivot.

So, in order to avoid ill conditioning and then, we proceed with rest of our operations. So, this the partial pivoting basically, requires that we choose r here, such that mode of a rk k is the maximum in that column, over k equal to one through n this term is the maximum in this column and after we find the maximum element in this column we interchange this row with our kth row.

So, this is known as partial pivoting, we also have something known as complete pivoting. In complete pivoting, we not only require interchange the rows, we also interchange the rows and columns instead of restricting to the kth column to find the element to interchange with the current a kk k the search is extended to other columns also.

So, a kk is replaced by the element with the largest magnitude in the partition basically let us look at our picture in the next slide. So, now, again suppose this is my near 0 element or 00 elements. So, instead of finding the maximum element in this column, we find the element which has the maximum magnitude in this entire partition, that is we look over all the rows and columns in this partition in the shaded region and find the maximum element and then, we interchange this column with that column and this row with that row.

So, basically instead of restricting to the kth column, we find the element to interchange with the current a kk k the search is extended to other columns also. So, extend the search to other columns and a kk k is replaced by the element with the largest magnitude in the entire partition. So, it is replaced by the largest element in the entire partition.

Thus complete pivoting requires that we choose r and s as this smallest integers, which satisfy a rs k is equal to maximum of a ij k, k is lesser than equal to i, j is lesser than equal to k and interchange the rows r and s and the columns k and s. Complete pivoting therefore, requires searching over the n minus k plus 1 elements of each of the n minus k plus 1 columns of the partition. To determine, the pivot thus complete pivoting is much more expensive than partial pivoting, because we have to search over all these elements in the entire partition.

In practice; however, partial pivoting is usually sufficient to prevent numerical instability and is commonly adopted. Thank you, we will continue with our discussion of Gaussian elimination in particular, we will look at some particular algorithms for doing Gaussian elimination for instance lu decomposition, crowd factorization, etcetera, which can further reduce the cost of Gaussian elimination by making at more efficient and more compact.