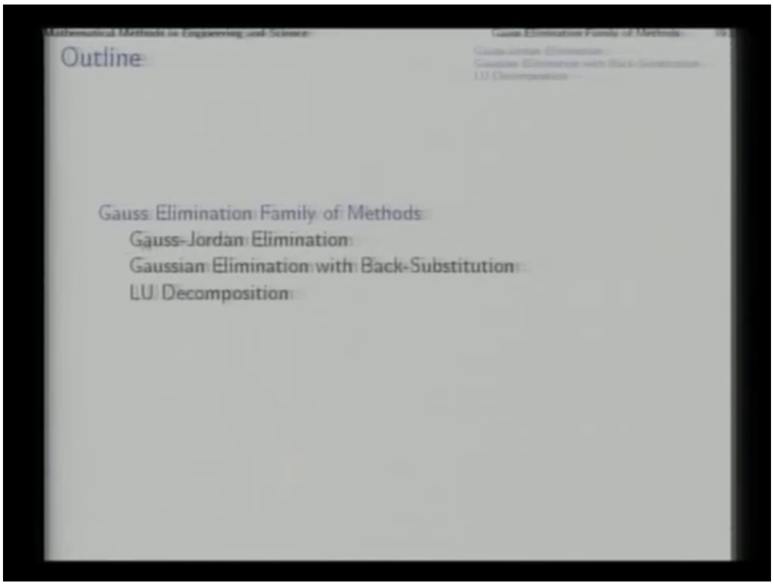
Mathematical Methods in Engineering and Science
Prof. Bhaskar Dasgupta
Department of Mechanical Engineering
Indian Institute of Technology, Kanpur

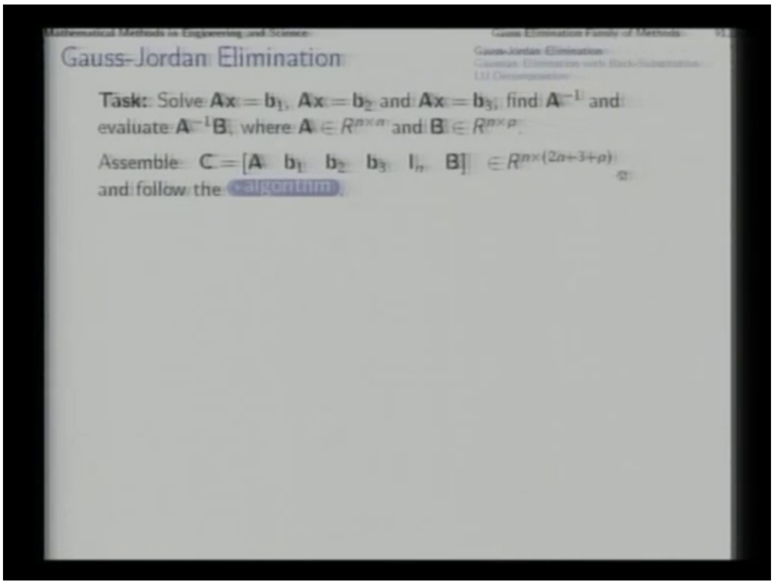
Module - I
Solution of Linear Systems
Lecture - 04
Square Non-Singular Systems

(Refer Slide Time: 00:26)



Good morning, today we will study the mainstream methods of solving linear systems of equations and in that we study these methods in the general family of Gauss elimination family of methods Gauss Jordan Gaussian elimination with back substitution and LU decomposition, these are the 3 methods that will study in this lecture. Now we have to note that these methods are typically applicable for square systems and the way, we will proceed is that we are expecting a matrix which is non singular as well, this is square invertible matrices; now when we have a system A is equal to B which you want to solve for known A and B.

(Refer Slide Time: 01:16)

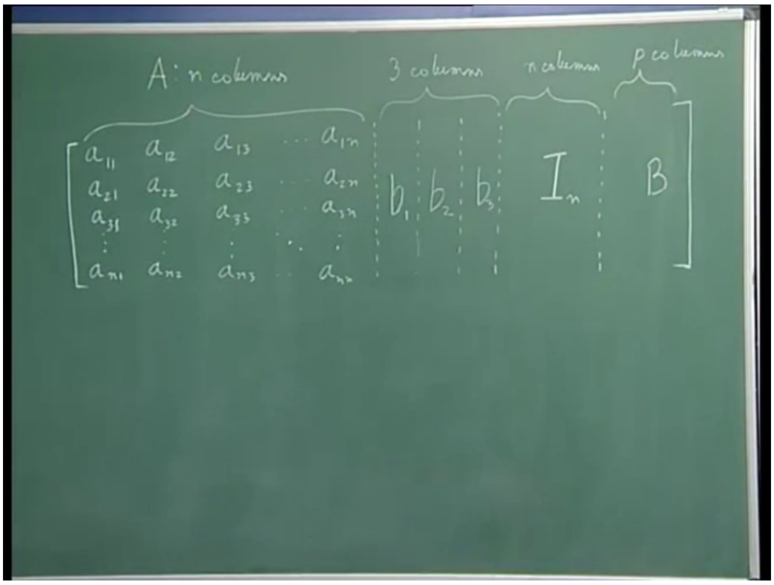


Then, first we see how we handle Gauss Jordan elimination; the good thing in Gauss Jordan elimination method is that the method goes in the systematic manner irrespective of how many right hand sides you have to solve. For example, if you have to solve for the system A x is equal to b for different right hand sides b 1, b 2, b 3 and at the same time, you want to find out the inverse of the matrix also which is same thing as solving the system A x is equal to b per n different right hand sides.

They are the n columns of the identity matrix because you know that A inverse is identity; that means, A into first column of A inverse equal to first column of identity matrix and so on and at same time, you can evaluate an expression like this which is A inverse B, this is again like solving a matrix equation that is A p equal to B. Now B is a known matrix. So, all these tasks we can do in the Gauss Jordan method in the same frame work.

Further first we assemble the matrix A; the right hand sides b 1, b 2, b 3 and an identity matrix if you want the inverse of A also and this matrix B here after assembling this matrix we have is a matrix of size n by 2 n plus 3 plus p in this 2 n plus 3 plus B p we have n columns.

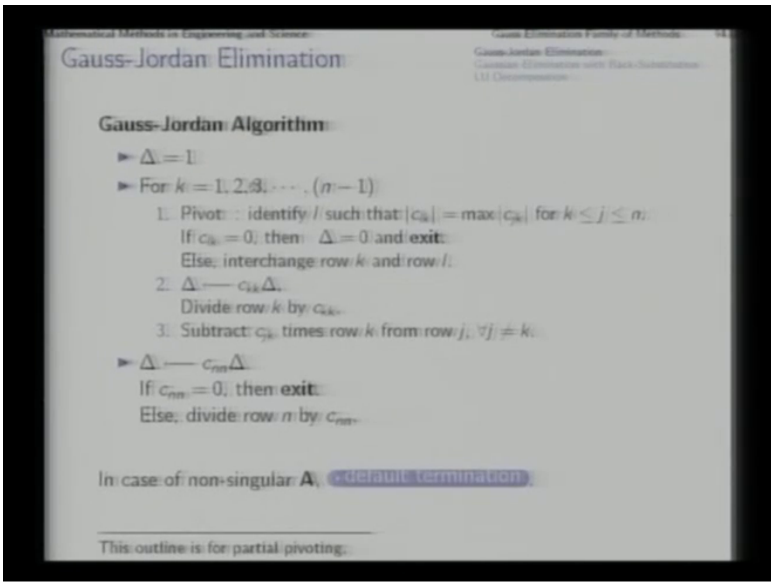
(Refer Slide Time: 02:55)



Here in A n columns; here in identity matrix 3 columns or the 3 right hand sides and 3 columns here in B, after we assemble this large matrix which has been denoted here as c, then we follow the algorithm the algorithm is actually very simple, what we have already studied.

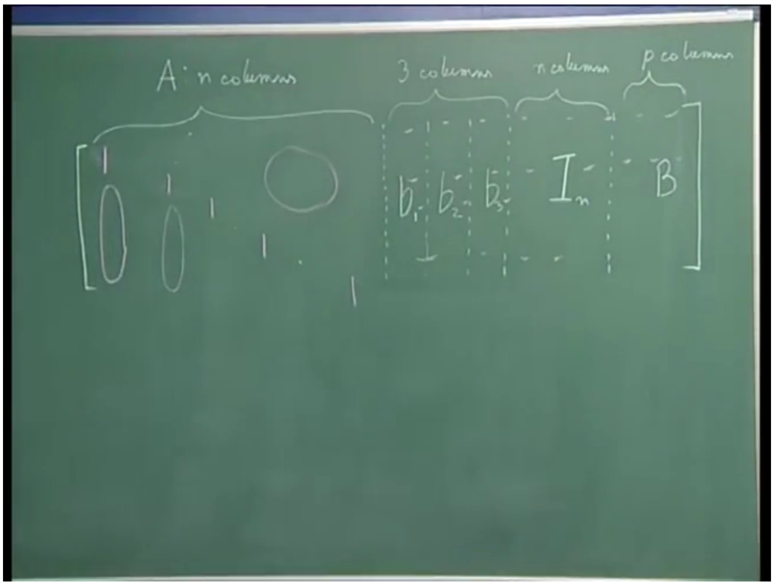
In the algorithm, we apply elementary row transformations to systematically reduce the matrix A to an identity; here, what we do first is that we want to reduce this matrix to identity matrix through elementary row transformations.

(Refer Slide Time: 03:27)



So, first thing we divide this first row by this diagonal element a 11 and then this becomes one.

(Refer Slide Time: 03:52)



This becomes 1 as I do this, all this undergo changes, right. Next we want 0s here in all these locations we want 0s, then we multiply the first row with a 21, a 31, etcetera and subtract from the second row third row and so on.

In the process, these fellows become 0 and these undergo changes whatever changes, we record them, next we want this diagonal entry to become 1, right. So, we divide this entire row, second row by a 22, as you do that this will become 1 and there will be further changes in the subsequent entries in the row. So, this becomes 1, next we multiply this row; second row with these entries and subtract from the appropriate rows. So, these again undergo changes and in the first time itself these fellows have changed and they keep on changing, right. So, now, here we have got 0 and so, on next we will chose this, right.

Now, in this entire process; if you find that there is a 0 somewhere which will mean that we cannot divide by that entry then we must pivot; that means, that from here to here we look for the entry with largest absolute value now; that means, that from here to here if you have 0 minus 0.12 minus 7 and so on, then in the largest entry minus 7 will be the because that has the largest absolute value. So, the sign plus or minus will not mattered we will take the largest value and accordingly we will make a row interchange now this pivoting is essential when a diagonal entry turns out to be 0.

But as a matter of policy as an algorithm it is done at every step even if the diagonal entry is not 0 so; that means, when you divide by the diagonal entry a k; k before that we do the pivoting step anyway now as we continue with this activity if the matrix is nonsingular if it is invertible, then by the end we will have all ones here right and note that when we apply the row transformation related to the second row; we get these as 0 and at the same time if we multiply the second row with a 12 and subtract from the first row this one will not get disturbed because it is already 0 here, but this will become 0 and so on.

That means that as we proceed for a nonsingular matrix; this whole thing finally, will break into an identity matrix about everything else is 0 and below also everything else is 0. So, this way, it will be by the time there will be a lot of changes here what will happen if the matrix is singular in that case in this n by n matrix as we proceed and we find that while pivoting we are filling at a diagonal entry which is 0 and below that everything else is also 0; in that case, we will find that from here to here, if the whole thing is 0 then we will not be able to pivot and we will not be able get a non-zero entry with which we can divide and that will be the fact for us to decide that this is a singular matrix and at that stage we can decide what we are suppose to do.

There may be an application in which you do not expect a singular matrix and in that case you can see the error and decide that this is a wrong data on the other hand there may be applications where a 0 has to be tolerated there you may try to make some other arrangement of handling the situation which will be discussing later. So, for the time being, we keep our attention focused on a square non singular matrix in which we get this identity; this is precisely the algorithm which you can see here the steps are exactly that pivoting an elimination and along with that there is one small extra item which is this delta.

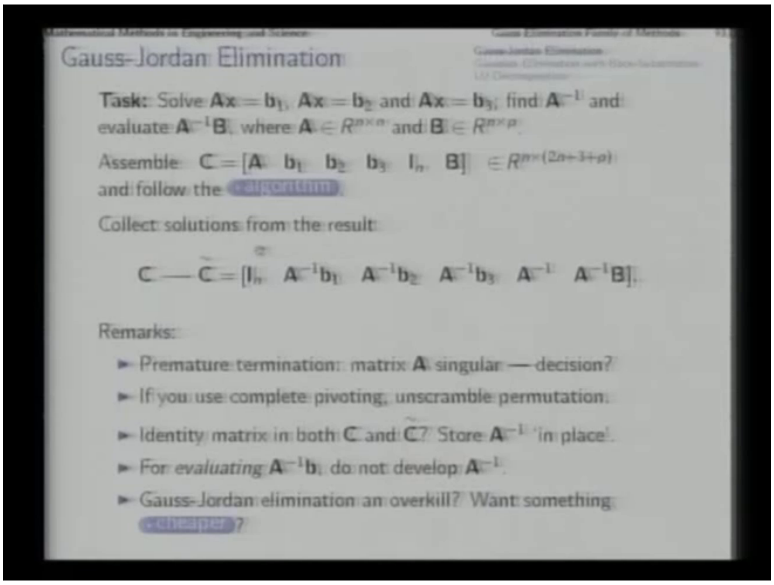
There are 3 lines which mention delta; delta equal to 1; initialization here if a pivoting process fails then set delta equal to 0 and x z that is a secondary situation where delta is used and a third is here that is delta update and there is another update here what is happening here is at this delta is actually the determinant. So, it is used in twos senses and one in order to use for the exit first that is if a singularity is detected then the determinant value is set to 0 and the program is exit it.

The other purpose for which the determinant is used here is by is for evaluating the value of the determinant itself that is as a byproduct that is this process of Gauss Jordan elimination in itself embodies a method to determine the determinant of a matrix for that as a separate routine is not required. So, in this method the determinant value turns up as a byproduct of the entire process. So, for that purpose first we initialize it and then further first second third diagonal entries we go on doing pivoting and the corresponding updates of the other rows. So, this pivoting step identify l such that absolute value of C l k is the maximum in the column from that diagonal entry downwards that is from k to n. So, when k is 3 we will be pivoting at this location.

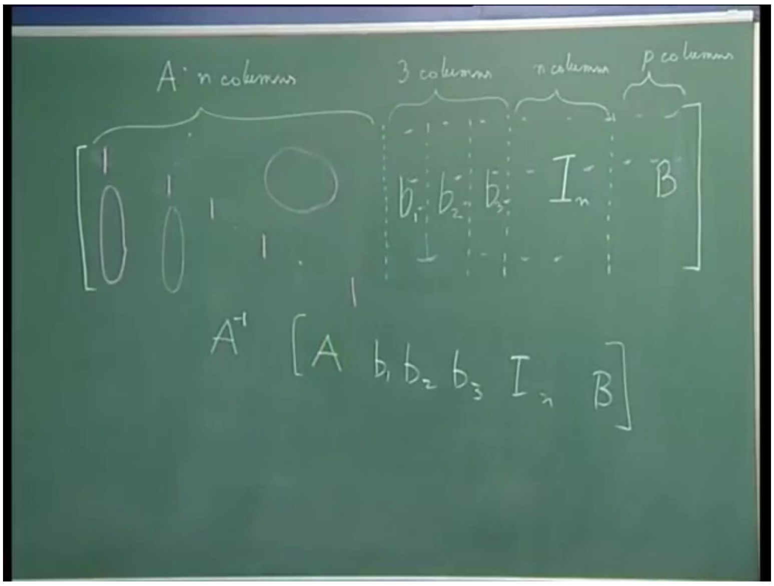
So, from here to here whatever is the largest absolute value entry that will be selected as the pivot and the k th row and that row l th row will be inter changed that is the pivoting step after the pivoting step you conduct the ordinary to elementary transformations dividing the diagonal dividing the current pivotal row by the diagonal entry C k k and then subtract the C j k times that row from other rows for all j above and below and that gives you 0s in the off diagonal entries above and below the pivotal point.

As you go on doing this then unless a singularities encountered you keep on doing till the end that is k equal to n minus 1 by the time you are through with k equal to n minus 1 everything has been reduced except that these entry till now may not be one. So, there will be a final need for one last rows scaling that is pivoting the last row by whatever entries sitting there right that is this step yes divide the row n by C n n before every such division of row by the current pivot entry current diagonal entry you need to update the determinant and that is why before division there is a determinant update here and there is a determinant update here.

(Refer Slide Time: 12:14)



(Refer Slide Time: 12:28)



At the end of the algorithm you have got determinant stored in delta and the matrix has been reduced to C tilde in which the first n columns will be identity and what will be there in the next columns note that we started with A b 1 b 2 b 3 I n and B now every elementary row transformation that we have applied is actually equivalent to a pre multiplication with a non singular matrix and that such matrices have been multiplied on this side many many of them, right and that will mean that all of them together will mean multiplication with one square matrix from this side and through that multiplication this square matrix has been reduced to identity.

Then you know which matrix has been pre multiplied that matrix is A inverse because A inverse is the matrix which multiplied with a will reduce this to identity we have applied equivalent row transformations here in order to reduce this to identity; that means, effectively we have multiplied this entire large matrix with A inverse and the result will be sitting in this matrix down and that will mean that by the time we have got identity here this will be A inverse b 1 there is solution of A x equal to b 1 similarly this similarly this and this will be A inverse into identity which means A inverse and this will be A inverse into b.

All the expressions that we need to evaluate all of them will be sitting here in the rest of the columns. Now when there is a premature termination that will mean that matrix A is singular and then you can decide what you want to do if you use complete pivoting now the outline that I just made is for partial pivoting if you want to conducted complete pivoting then for every column inter change the variables will get scrambled and; that means, you need to store that permutation and then later after solution you need to unscrambled that permutation of the xs; x 1, x 2, x 3, etcetera.

In our opinion that is not really necessary in most of the applications you can carry though the with the partial pivoting list algorithm as outline here now if you issues we can raise regarding efficiency in this native algorithm you would have noticed that in the beginning here we have stored an identity matrix which was not needed because it does not have any information content as long as we know that an identity matrix is supposed to be there we do not need any further information to actually store those 1s and 0s similarly at the end, we have got another matrix sitting here which is again not necessary to really stored.

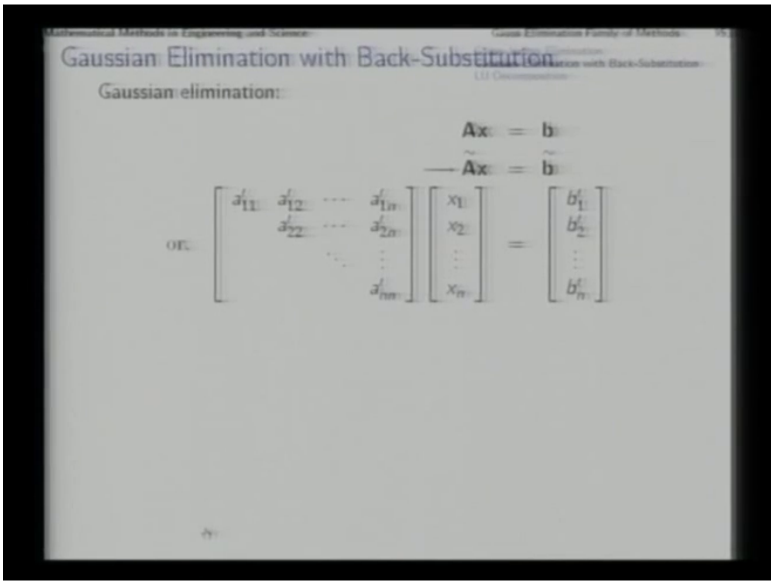
Therefore many of the numerical routines professional algorithms do not give extra space for this and here for this what they do is that whatever will be the result of the calculations in these locations that are stored in the location of a itself in the process now if you are new in this kind of algorithm calculations computations, then I would not advice you to implement that, but then numerical algorithms professional algorithms can do that because after the after a particular entry here has been obtained in its final form then later you will never need the old one.

Therefore it is not a problem to override the old number and that this fact you should know because in your actual research problems at many occasions you may actually use numerical algorithms and the way they handle your data and the way the report your result you should know very well and therefore, when you call a subroutine from any professional library, then you should make sure what the algorithm for the implementation of this algorithm in the library does to a if the original matrix A is getting over written by A inverse finally, this one.

Then it is important on your part to store the original A by A different name before sending that matrix for to the library another important point is that many students while solving for A x equal to B 10 th to first find the inverse of a and then multiply A inverse B that is not a good idea for evaluating A inverse B never develop the complete inverse because developing the complete inverse will actually require you to solve A x equal to B kind of systems for n different bs that is n columns of the identity matrix.

Why should you spend so, much of computationally effort to just solve one system for a single B therefore, for evaluating this for solving a single B you should not actually develop the complete inverse in that case you can just curtail your system to this point and apply the algorithm that will be much cheaper now you might find that if you write the algorithm or if you try to do an hand calculation then you will find that Gauss Jordan elimination this algorithm is perhaps and work is regarding number of computational steps additional subtraction multiplication etcetera if you want something cheaper, then the option is the second algorithm of this family and that is Gaussian elimination with back substitution.

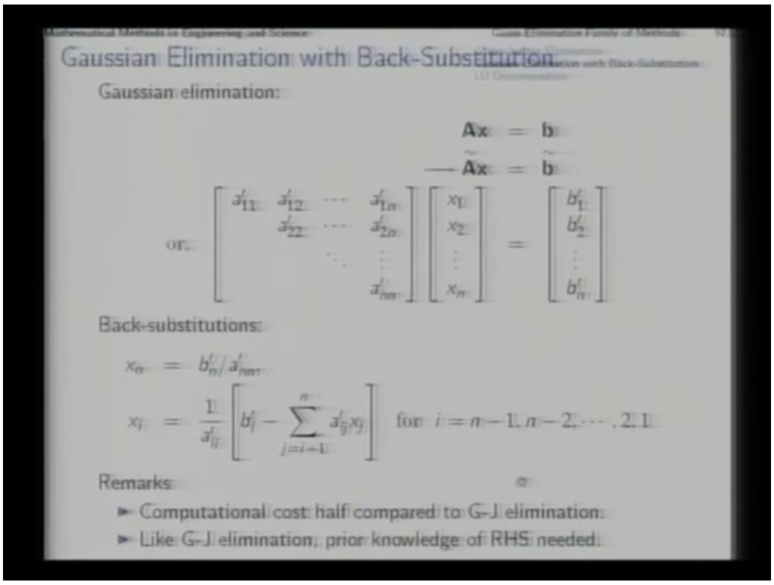
(Refer Slide Time: 18:18)



You might say that this complete reduction of the matrix A up to identity through elementary row transformations is actually not because as long as we can reduce it to an upper triangular form in this manner that is below this everything else is 0 our job is done as the if we can just apply elementary row transformations. So, that a reduces up to the stage of a tilde where a tilde is this matrix with upper triangular structure below that below the diagonal entries everything else is 0 then our job is actually done because here we can use the last equation first and determine x n because in the last row we had everything 0 except the last entry.

That means this last equation opened up will give us a nn prime x n equal to b n prime; that means, x n is directly determined then we can handle the equation just above that will have 2 unknowns x n minus 1 and x n; x n is already known then x n minus 1 can be determined that is one at a time, we can determined if we come from the lower and upwards that is called back substitution this is why we say that this method is called Gaussian elimination up to this point and then back substitution.

(Refer Slide Time: 19:49)



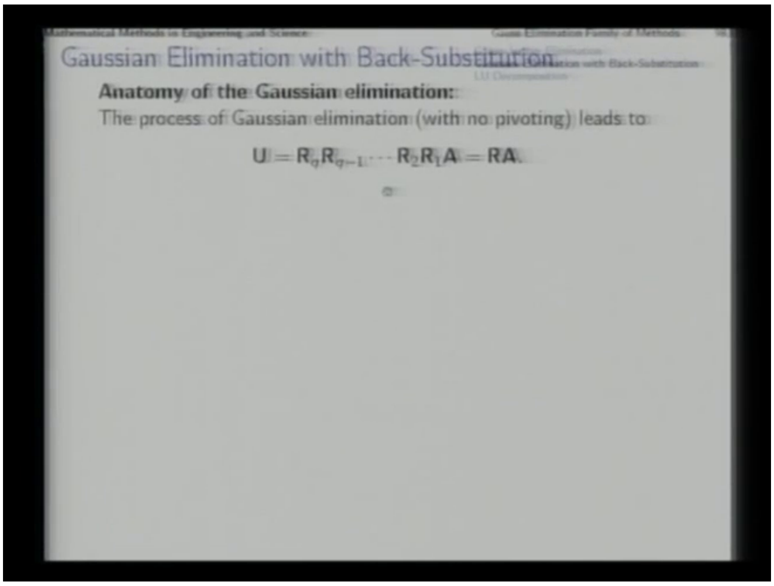
So, this we can do and the last unknown is determined first and then x n minus 1 is determined first I equal to n minus 1 is used and then after x n and x n minus 1 is determined then the previous one then the previous one; I am through one complete series of back substitutions you determine all this variables.

So, this method Gaussian elimination with back substitution is somewhat cheaper than the earlier method you can say that the computational cost for a single right hand side is actually half compare to Gauss Jordan elimination method which we discussed first; however, if you need to solve the system for several right hand sides large number of right hand sides or if you want to find the inverse also is that is needed then the Gaussian elimination will not offer you any advantage; that means, between these 2 methods if you want to solve the systems for several right hand sides or if you want to find the inverse also in that case you should prefer Gauss Jordan elimination.

On the other hand, if you want to solve it for a single right hand side like is then you should prefer this method, but in both of these methods prior knowledge of RHS is needed prior knowledge of the vector B is needed because the elementary row transformations which is applied to a at the same time is also applied to the right hand side now you might find that another algorithm which will not need the right hand side beforehand, but will first process with the matrix A itself and keep it in such a ready form that the moment a right hand side vector appears you can do another little bit of calculations to find the solution.

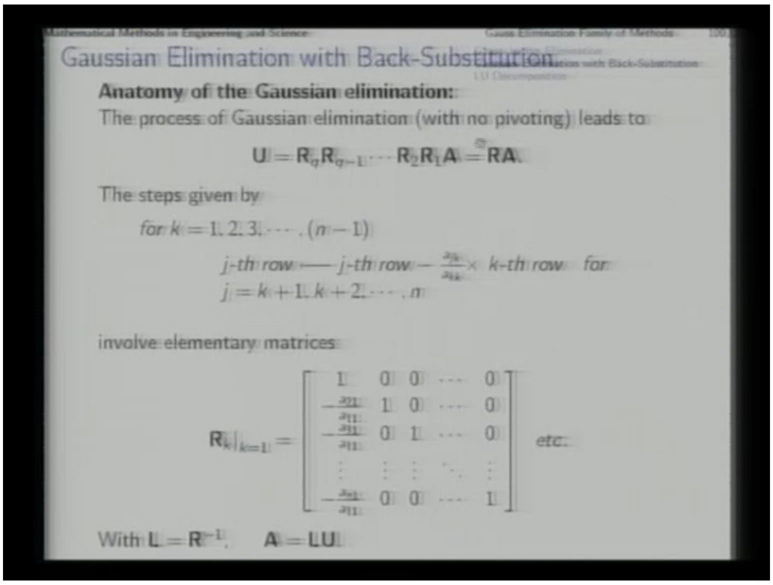
It is like what they do in good restaurant they keep the arrangement ready for a large number dishes and the moment the customer orders something that dish is prepared and brought by. So, that kind of a situation emerges when you study LU d composition and LU d composition also is a member of this particular family or family of Gaussian elimination methods and before going into the actual method as such we will try to make a little observation regarding how this Gaussian elimination method actually work what are the steps that were applied in order to reduce this matrix A to a tilde this upper triangular form.

(Refer Slide Time: 22:31)



So, when you try to explore the anatomy of this Gaussian elimination process then we found that the matrix A was sequentially operated over by a large number of elementary row transformations equivalent to pre-multiplication is corresponding elementary matrices R 1, R ,2 R 3, R 5, etcetera the result was a matrix U.

(Refer Slide Time: 22:59)



Which is upper triangular the steps where this in which; now if you do not want the diagonal 1, then what we need to do is at j th row is updated by subtracting from it a j k by a k k into the k th row right.

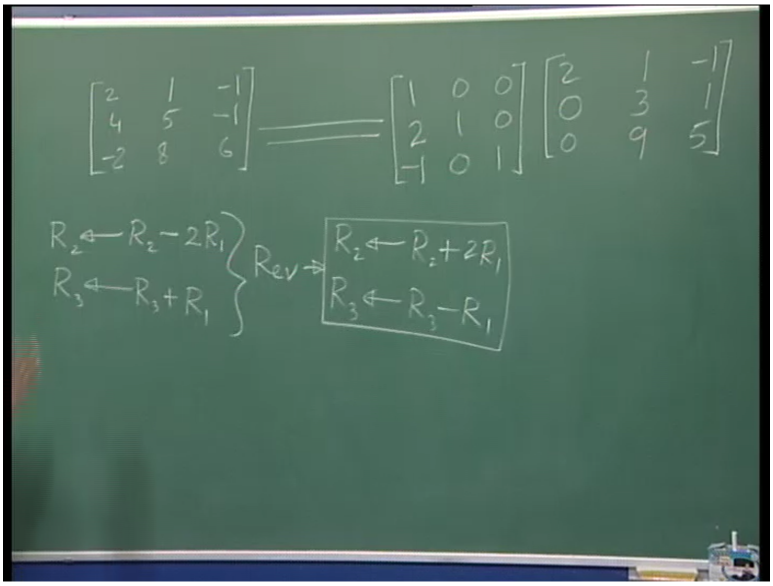
For j varying from k plus 1 up to the bottom that is n and that will mean that say in the first case for k equal to one what we will be needed we will be conducting the second row minus a 21 by a 11 into the first row that is equivalent to having an entry here right like this and so on. So, these step for a running from j running from k plus 1 to n is actually going to this right now. This is the matrix R 1 with which the matrix A was pre multiplied during the step in which who you would get all 0s here without bothering to get a 1 here and so on.

Similarly, in the next case when we want do reduce the a 32 to a n 2 these things to 0, then we will be have having another set of elementary row operations in which the R 2 matrix will emerge as the corresponding elementary matrix in which similar entries will be found here everything else is same as identity matrix right and so on. Now you will notice that this matrix and in the next matrix in which these fellows will have entries like this and so on; all of them are lower triangular matrices and there is something very interesting in lower triangular matrices.

If you have a large number of lower triangular matrices here then the product is also a lower triangular matrix; that means, this entire matrix product R is a lower triangular matrix and there is a further interesting point here that the inverse of a lower triangular matrix is also a lower triangular matrix this is one of the exercises in this chapter in the book which I strongly advise you to do on your own. So, that you appreciate what is happening there and how that appears for example, the inverse of this matrix is exactly this matrix itself with all these minus signs becoming plus and nothing else.

This structure of the inverse of lower triangular matrices with the original matrix has a relationship with elementary row transformations and their inverses which we will see now if this matrix R is lower triangular and its inverse is also lower triangular then calling that inverse as l you can see that LU will become l R A and R is A; L is R inverse; that means, LU is a this is the basic idea behind the algorithm for LU decomposition; let us understand this let us try to appreciate this whole thing with a little example suppose we want to conduct a Gaussian elimination of this matrix, we will do that we will apply the Gaussian elimination at the same time we will do a few more things we will analyze.

(Refer Slide Time: 26:20)



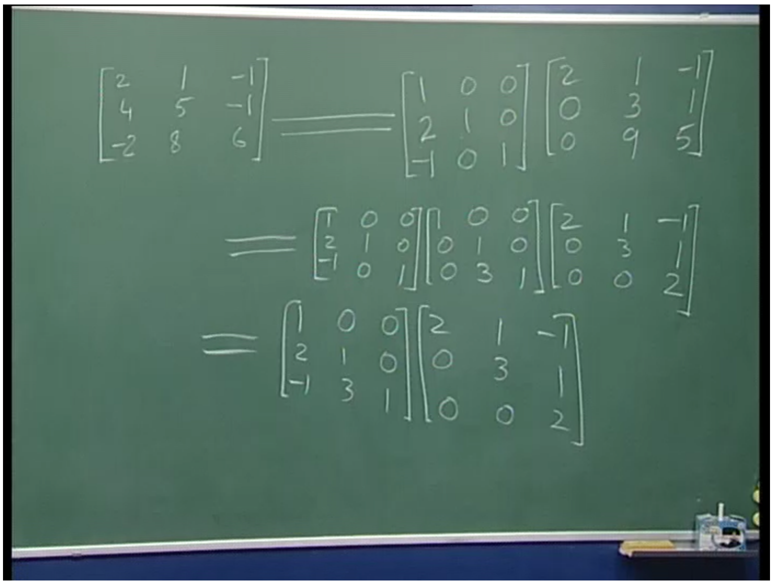
How this Gaussian elimination takes places in short we will be actually studying this anatomy of the Gaussian elimination. So, first round while conducting the Gaussian elimination what you want to do you want to get 0s, here in Gaussian elimination itself you do not bother to get one here you just one 0s in the sub diagonal locations. So, these 2 locations you want to make 0. So, what you will do you will say said from the second row I will subtract twice the first row and to the third row I will add in the first row that will give me 0s here let us do it.

The result will be first row will remain unchanged from the second row we are subtracting twice the first row; that means, we will get 0 31 to the third row we are just adding the top row first row; that means, you will get 0, 9 and 5; now tell what will you do if our mood changed we have lost this matrix, suppose; we have lost this matrix, but we have go this and in our hand, we have the information regarding what elementary row transformations were conducted to get this and now we want to regain back the original matrix.

To get that original matrix what we need to do what we did earlier from the second row we subtracted twice the first row and from the third row we actually added the to the third row we actually added the first row. Now we in order to regain the original matrix what we can do see this first row was unchanged. So, the this is the old first row sitting here that we have not lost; that means, if you add back twice the first row here and if you subtract this top row form the third row then we should be able to get the original matrix; that means, the reverse operation of this if you want to reverse it that will be R 2 should get back its lost part of twice R 1.

Similarly, R 3 should get updated with this that will be restoring the original matrix right. So, what is the corresponding elementary matrix for this the elementary matrix for this is 0 0 0 that identity matrix changed through these row operations; that means, we will get 2 here and minus 1 here right; that means, through these elementary row operations or through pre multiplication with this elementary matrix equivalently we will be able to regain the original matrix.

(Refer Slide Time: 30:20)



So, we can write the right now you can forget about this part now you have got 2 things in our hand; one is a little reduced version of this matrix sitting here and another matrix note it is low triangular another matrix which pre multiplied with this will actually give us the old matrix back. Now to get it completely in the upper triangular form, we want a 0 here; that means, we can subtract from this last row 3 times this we should not touch the first row because that will spoils this 0 right because there is a 2 sitting here.

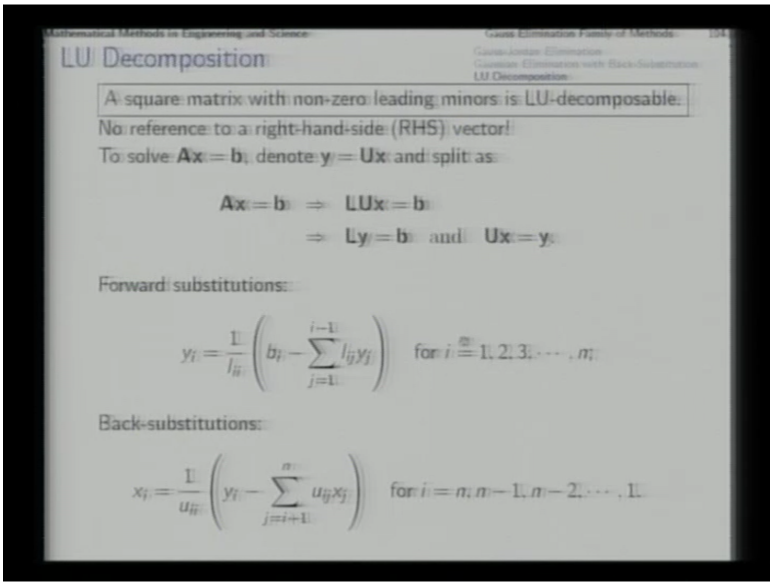
This is 0. So, it is not a problem. So, from the third row 3 times the second row, we will subtract to get a 0 here. So, 3 times the second row subtracted from the third row will give us this, right, thrice, the second row has been subtracted from the third row. Now again, what we did we apply this elementary row transformation. Now again, we want to get back the original matrix. So, original matrix, we will get back by this operation, right. So, this remains in its place that it be there and we want to get back this matrix by inserting something here, right.

So, what is that thing that matrix; we will be the elementary matrix corresponding to this elementary row transformation and that will be that is the easy to get that is through addition of thrice the second row of this to the third row of identity, right. So, now, we will see another low triangular matrix. So, this is again equal because this is same as this and this is the same as the product of this; now you can forget about this and this whole thing is equal.

The product of this 2 lower triangular matrices will give us a lower triangular matrix which through multiplication you should verify. So, this is the decomposition of the original matrix into 2 parts 2 factors one is lower triangular and the other is upper triangular right and now what you can do is that now you call the right hand side till now there has been no talk of the right hand side and we have come up to this point this decomposition is called LU decomposition; that means, not necessary to apply the actual elementary row transformations on A to the upper triangular form up to this form.

But separate out from the matrix A, those elementary row transformations in the form of this lower triangular matrix such that the remaining thing remains upper triangular. So, here you have collected the set of elementary row transformations in the low triangular matrix and on that side the upper triangular matrix is remaining. So, this is the spirit of LU decomposition. Now note that in this, we have taken such an example in which pivoting was never necessary and that is why here in the beginning we discuss the process of Gaussian elimination for those matrices for which no pivoting is there.

(Refer Slide Time: 34:50)



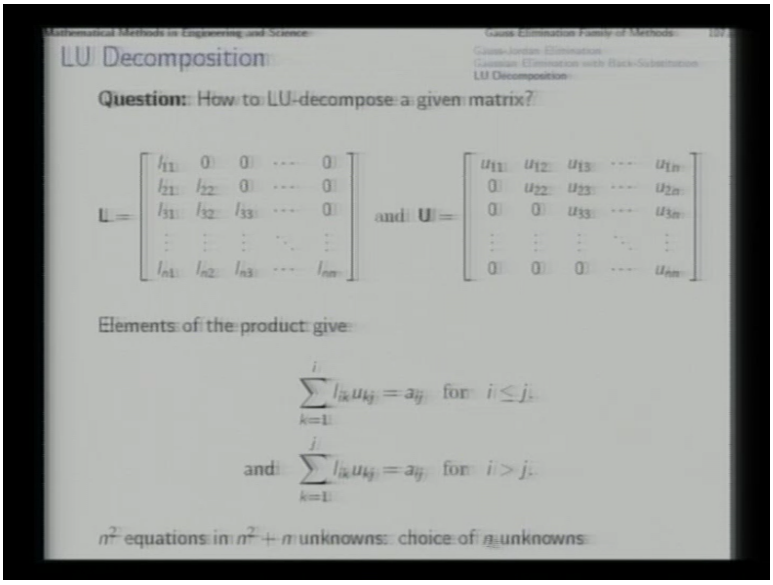
This is not a serious limitation in the actual process, this was done in order to explain the situation easily now in the method of LU decomposition the most important underlying understanding is that the square matrix when process through such elementary row operations does not have does not acquired a 0 in the diagonal position such matrices will have all minors leading minors non-zero; what are the leading minors as case; this is the first leading minor the second leading minor is the determinant of this the third leading minor is the determinant of this and so on.

For an n by 1; n by n matrix, there will be n such leading minors all those determinants should be non-zero for a matrix to the LU decomposable; in this manner, if suppose such is the case for our matrix at hand, then how do you proceed in the preceding, we will not make any reference to the right hand side vector, but suppose, we have decomposed it like this and then we are supplied with a right hand side vector, then how to solve it that solution is very easy to solve A x is equal to B, we will denote U into x as y and then a decomposed in this manner l lower triangular u upper triangular LU.

Then in place of a if you write LU, then Ux represented as y will give us l y equal to B and then Ux is y that is what we have denoted now since B is known and the lower triangular matrix is known why can be immediately determined to a process of forward substitutions because if this kind of a matrix LU l matrix is sitting as the coefficient matrix then the first equation we will have only x 1 which you can determine the second equation we will have x 1 and x 2 after determination of x 1 x 2 can be determined and so on. So, these are forward substitutions. So, through a series of forward substitutions like this we determine the complete y; y equal to 1 to n after y is determined, then we address this system in which u is upper triangular like this which is our old friend and in that case we conduct a series of back substitutions.

So, which is a series of forward substitutions followed by a series of back substitutions we can solve the system of equations for every right hand side it once we have converted the original matrix A into l and u factors now the question still remains how to conduct the LU decomposition process you certainly would not go in this manner because this will be extremely costly because this here we have actually done the done double the job right in order to examine the anatomy of this entire process now well decomposing a matrix into L and U factors, we first recognize that the factors will look like this and these are the unknowns which we will need to determine.

(Refer Slide Time: 37:59)

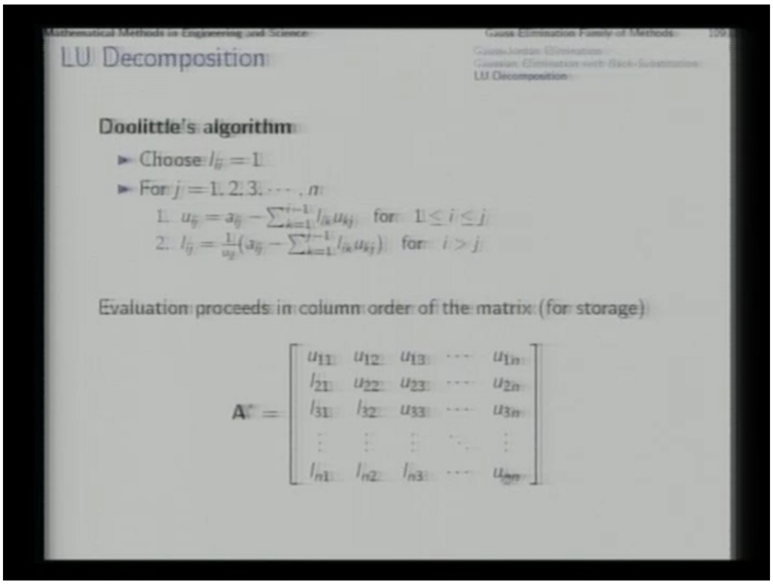


The number of unknowns here, now these are 0s the number of unknowns in this first matrix is n into n plus 1 by 2 exactly that many unknowns here right so; that means, and if you work out the products then products will look like this normal matrix multiplication because LU together after multiplication should give you a; that means, a i j is the product of i th row of l and the j th column of u that has been written here in 2 ways because only f u of the trans you will need to add not from k equal to 1 to n, but k equal to 1 to i because beyond the diagonal entry we will not need.

So, when i is less or equal to j you can use this i terms as summed up when j is less, then j terms can be summed up, right. So, now, you find that n; n plus 1 by 2 and n n plus 1 by 2 unknowns here and here a total number of unknowns is n into n plus 1; that means, this many unknowns and how many equations LU equal to a. So, a has n squared elements. So, when you equate term by term then you get n square equations. So, less number of equations in more number of unknowns; that means, n extra unknowns are there that many unknowns can be chosen.

So, one particular choice leads to the famous algorithms called Doolittle's algorithm and that is by choosing the diagonal entries of the lower triangular factors as one all these diagonal entries are chosen as one and the others other n square entries are determined and that determination can be done in the particular order such that the at every step you will be solving one simple equation in one unknown only.

(Refer Slide Time: 40:14)



So, after choosing L i, what is done here it is written in algorithmic form which we should later verify, but currently look at this.

After determining after assuming l 11 to be 1, you should determine u 11 in this order in this order you should try to determine first determine u 11, then l 21, l 31, l 41 and so on, then u 12, u 22 and then the sub diagonal entries from L, then super diagonal and diagonal entry of the third column of u and then the sub diagonal entries of L below the third entry which is known to be 1 and so on; if you conduct it in this manner, then you will find that at every step you are solving just one equation in 1 and 1.

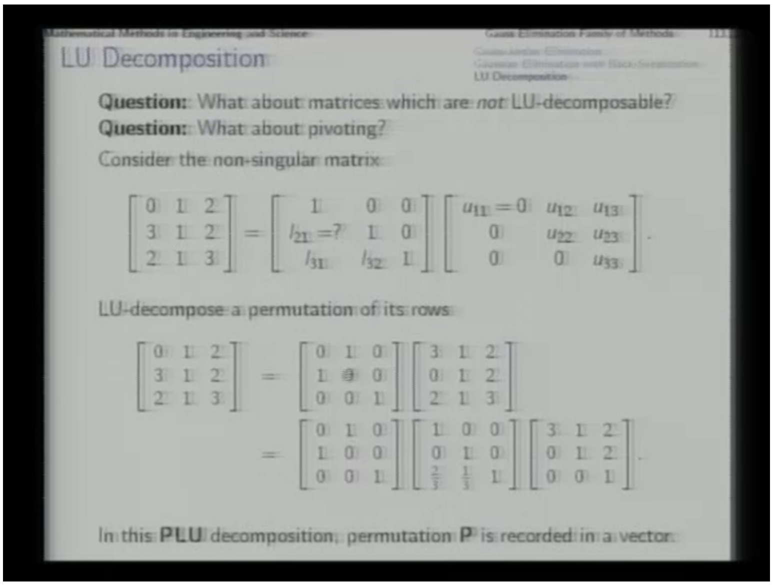
See, here a 11 when you try to solve you get l 11 which is 11 into u 11 and nothing else; that means, a 11 will come here then you try to solve for this. So, second row into first column l 21 into u 11 is equal to a 21 and u 11 is already determined. So, you get this. Similarly, you get all these, next you will determine these 2 and this is already 1, then you will determine all these. So, when you write the equations in this order, then at every stage, you will be determining 1 and 1 u 11; all these this 2; then all these, then these 3 then all these and so on.

In this order that is the order is actually written here, you can determine all these unknowns one by one and most of the professional algorithms for LU decomposition actually give you the output in this manner in which this matrix as such as no value except that the sub diagonal entries list out the non trivial entries of l others are ones and

0s and the diagonal and super diagonal entries give the appropriate non trivial entries of u because in that the sub diagonal entries are all 0. So, this saves storage space.

Many professional algorithms also make another economical use of storage in the sense that this matrix the written in the place of a itself the address the location in which u supplied a in that name in that place itself it sense you back this matrix. Now this is do it these algorithm.

(Refer Slide Time: 42:58)



Now we address that original question which we avoided in the beginning what about matrices which are not LU decomposable at the same time the other question arises what about pivoting for example, take this; this matrix is non singular; that means, that we should not be saying that any Gaussian elimination method will fail because it is a singular and the diagonal entries 0, etcetera, etcetera.

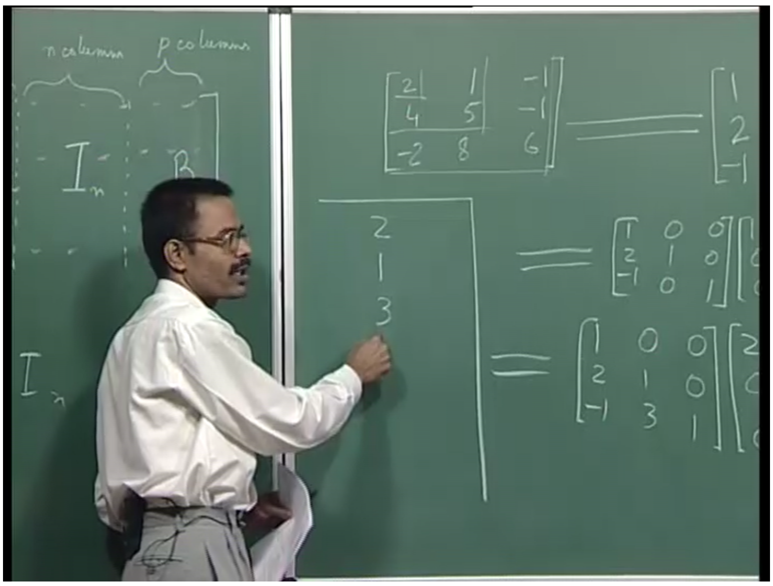
We should not be saying that; that means that for this non singular matrix we should be able to apply any Gaussian elimination terminology of method without any hindrance. Now let us proceed as we proceed first we try to determine what is the u 11 entry because l 11, l 22, l 33 they have been already that as one above that everything else is 0 all sub diagonal entries of u are already 0. Now we want to first determine this u 11. So, for that we equate a 11 with the product of first row of l and first column of u; that means, one into u 11 plus 0 plus 0; that means, u 11 is this.

That is 0, we have found it next we will determining l 21 and l 31, but the moment we tend to determine l 21, we need to write the equation for a 21, a 21 is second row of l into first row of u; that means, l 21 into 0 that is 0 plus 1 into 0 that is also 0 plus 0 into 0 that is also 0; that means, in this product all 3 entries are 0 so; that means, the sum is 0, but here it is 3 and we are stuck.

We are stuck because of this diagonal entry which is 0 which should not be sitting there, but it is sitting there what you can do. So, for this kind of situation we must do pivoting, but then when we do pivoting what happens the row structure changes. So, what we do is at in LU decomposition; whenever we do pivoting in actual practice we do it at every step. So, in this particular case while pivoting we will be bringing the second row at the top. So, when we conduct that row interchange between first and second row then we get this matrix.

But then we do not know the right hand side later when the right hand side will be given to us at the time we should know that in the right hand side vector also the first and second row should be swept for that purpose we keep track of the permutation see this is permutation this can be stored not necessarily in a matrix, but in a vector and that vector is 21 3; that means, second row of identity matrix first row of identity matrix third row of identity matrix 21 3 and this is that complete matrix.

(Refer Slide Time: 45:51)



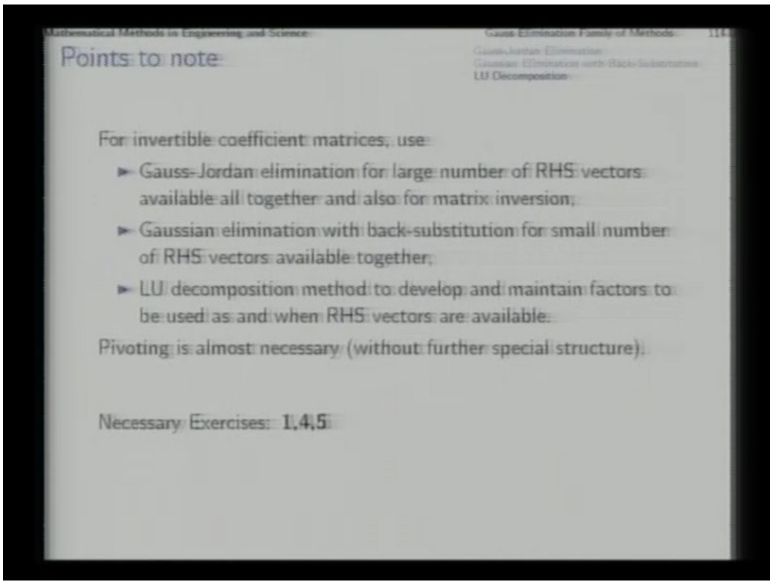
Second row of identity matrix first row of identity matrix and third row of identity matrix; so, this is a matrix which is called permutation matrix corresponding permutation vector is this; that means, we can conduct LU decomposition with pivoting in that case basically, we will not be decomposing the original matrix given, but we will be decomposing a permutation of it rows and what permutation that we will be separately storing in a vector like this which is equivalent to stored this permutation matrix. So, now, with this permutation matrix sitting here we have got this matrix which is a permutation of its rows and then this can be easily decompose into l and u factors which are these which you can verify later that the product indeed gives you this matrix.

That means that rather than conducting an LU decomposition we actually conducted a PLU decomposition in which the matrix P is a permutation matrix corresponding to which we stored the permutation vector and later when right hand side vector appears for which we need the solution then we accordingly permute the entries of the right hand side vector also and then conduct the rest of forward and back substitutions now you will ask the question why are we stressing this point that the right hand side vector will be given to us later.

If so, then can we not conduct the whole operation later on the point is that there are many many applications; many many computations in which a matrix stays as it is and at different times at different stages of the entire large computation several right hand side vectors appear for which we need to find out the solution many times we cannot set up the right hand side vector until we solve the same system before hand for another right hand side vector.

In such a situation where the same matrix A will be needed for solution of several right hand side vectors at different stages of the entire large algorithm it helps us a lot, if we conduct the LU decomposition once and then stored that L and U factors and then whenever new right hand side vector emerges in possibly subsequent iterations, we conduct the forward back substitutions this gives us efficiency of the entire process

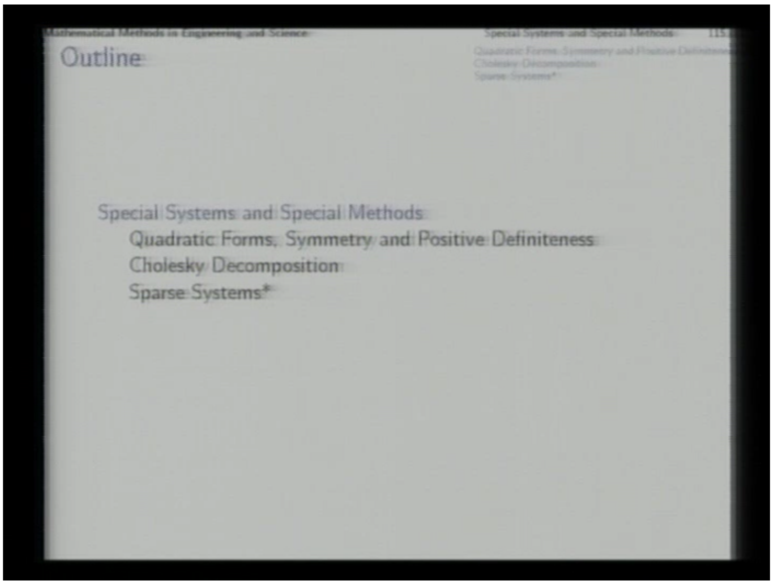
(Refer Slide Time: 48:48)



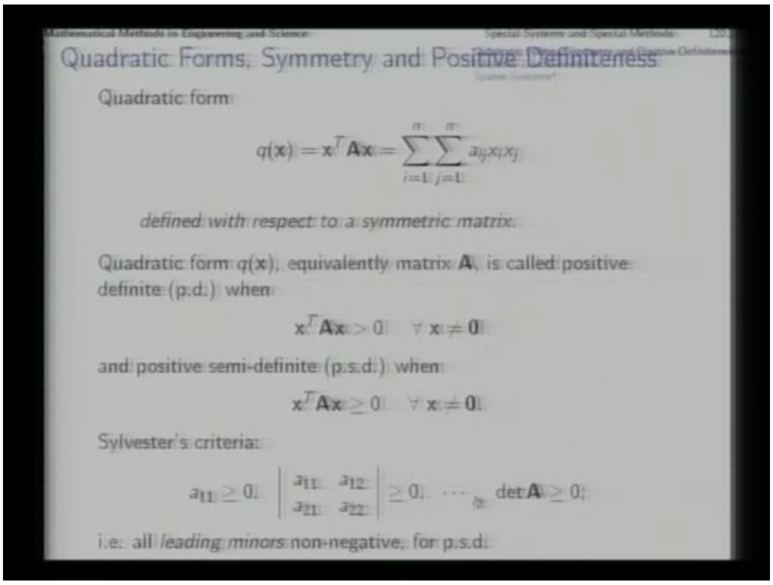
Now in this lesson, till now, we have covered the method called solving linear systems of equations with square non singular coefficient matrices and if by accident a singular matrix appears there, then we know how to determine the singularity how to detect that singularity and the other case in which infinite solutions etcetera would be possible that we have already covered in the previous lecture.

Two more important issues come into practice one is that when the given system given square matrix is particularly good then how to take advantage of that particular good feature and when the given matrix A is particularly bad in that case how to handle the bad features of that matrix particularly good what we mean by that; by that we mean special systems for which special methods we will take advantage of the particular advantages situation before introducing a particular special situation we quickly see if a few definitions.

(Refer Slide Time: 49:46)



(Refer Slide Time: 50:00)

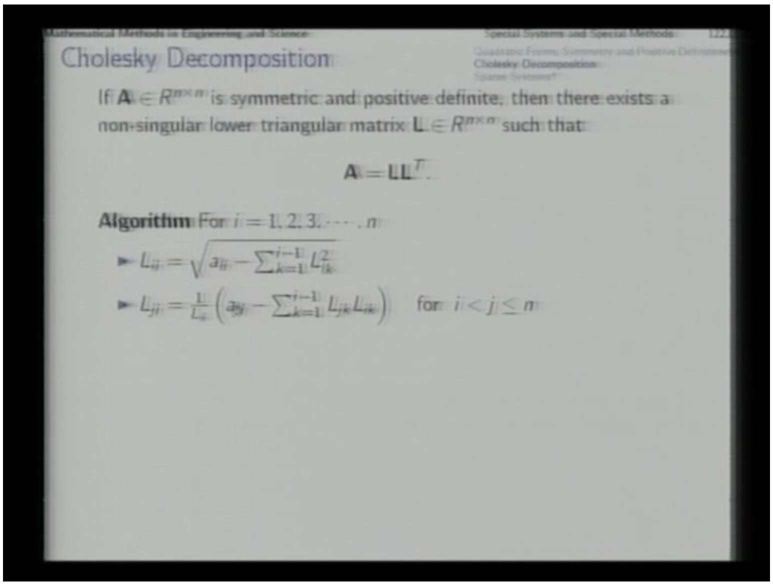


A function of this form quadratic function of x 1, x 2, x 3, x 4 up to x n in which only quadratic terms are there, no constant term, no linear term is called a quadratic form which can be shown as x transpose A x which is this in the summation notation. Now this kind of a quadratic form is typically defined with respect to a symmetric matrix A because in any case, if you try to put a non symmetric matrix there one can quickly change for example, if a were not symmetric then in place of a we put a plus a transposed by 2 giving the same function, but in a matrix which is symmetric that is why this kind of a quadratic form which is typically defined in terms of a symmetric matrix A such matrices appeared enormously in practice in science and engineering.

For example the stress tensor inertia tensor all of these are represented with symmetric matrices. So, this is always defined with respect to a symmetric matrix now the important definition which we should keep in mind is that this quadratic form this kind of a function and equivalently the corresponding underline matrix A is called positive definite when this evaluates to positive values for all non-zero x and for 0 x obvious it is 0, it is called positive semi definite if the value of this evaluates to non negative positive or 0 for all non-zero x. So, these 2 definitions of positive definiteness and positive semi definiteness we should keep in mind.

Now, there is a test by which you can determine whether a matrix is positive definite or not this is for positive semi definiteness when you remove this less a greater than or equal to signs with strict inequality then you get that is for positive definiteness this is called Sylvester's criteria, but as you know it will become quite possibly computationary because you may need to remain a lot of determinants anyway if a matrix is positive definite then a linear system A x equal to B with that matrix A as the coefficient matrix is cheaper to solve and that method is colicky decomposition.

(Refer Slide Time: 52:47)



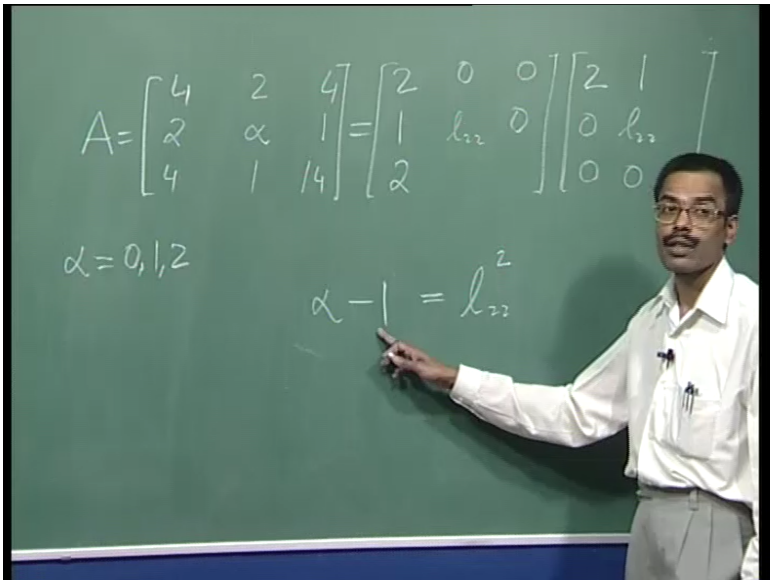
So, if the matrix A n by n matrix A symmetric and positive definite then there exists say non singular lower triangular matrix l such that is happens what is here being done in a

Doolittle’s algorithm for LU decomposition we made a choice regarding the diagonal entries of l we said that all of those we want to the ones here we make another choice for the diagonal entries rather we make a choice in the nature of the l and u factors we say that we want l and LU decomposition in which the u factor and l factor are transposes of each other and that you can top up when the matrix A is symmetric because l l transpose will always give a symmetric matrix.

So, is the matrix A is symmetric then you can ask for an LU decomposition in this manner in which l is lower triangular and the upper triangular factor u is its exact transpose now when the matrix A is symmetric you can talk of such a decomposition and when you try to conduct that decomposition you will succeed when the matrix A is not only symmetric, but also positive definite. So, for that algorithm is actually the same as or similar as LU decomposition algorithm in which one by one you try to find out l 11, l 21, l 31, then l 22, l 32, l 42 and so on because above the diagonal terms you do not need to find because the u factor is actually the transpose of the L factor and you need not find it separately.

The rest of the algorithm for the solution of the linear system is same as in LU decomposition through A series a forward and back substitutions. So, this algorithm which has been elaborated in pseudo code in the slide here you should verify later currently what we do we try to see that through an example in which we also try to see why the matrix should be positive definite; suppose the matrix is A with these entries 4, 2, 4, 2 alpha 1, 4, 1, 14.

(Refer Slide Time: 55:05)



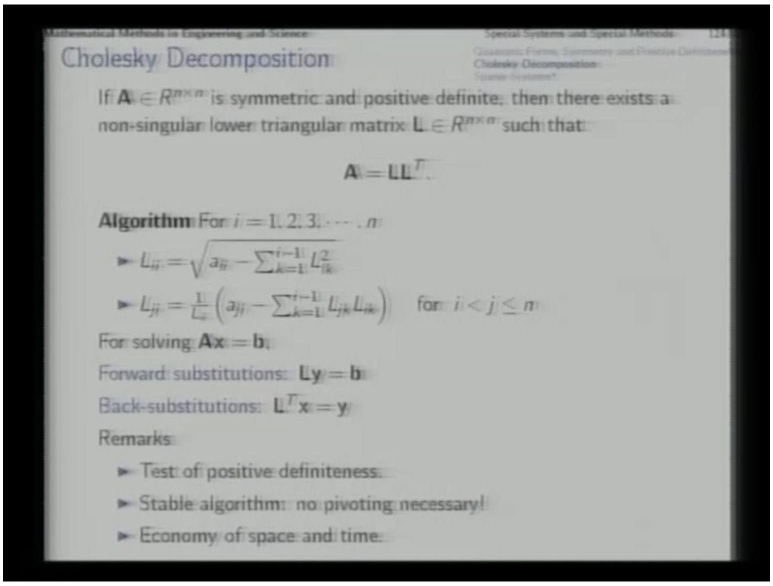
We want to decompose it in one and one transpose factors with these 0s already known that for 3 different values of alpha 0, 1 and 2 first what should be this value l 11 which is always sitting here because this is just transpose of this. So, for finding that we try to equate the a 11 entry on both sides. So, 4 is equal to l 11 into l 11 plus 0 plus 0; that means, l 11 square is 4. So, what is l 11, we take the positive entry; next we try to find these entry l 21 and for that we equate this term to the corresponding of this time.

So, 2 is equal to l 21 into 2 plus something into 0 plus something into 0 so; that means, l 21 is 1. So, we get one here one here next for this term we say that will have something into 2 plus 0 plus 0. So, that something should be 2 we get this correctly for this we do not need to do any computation because this is symmetry for alpha this is the crucial step we have here alpha equal to 1 into 1 plus l 22 into l 22 plus 0.

Now, for alpha equal to 0, we will have l 22 square is equal to 1. So, you can write it like this first for alpha equal to 0, we will have l 22 square equal to 1 minus 1; that means, that it is not be possible because we want the factors to be real for l 22 for alpha equal to one will have l 22 equal to 0 which will be find at this stage, but at the next stage we will need to divide by that l 22 and we will be stuck there on the other hand for alpha equal to 2 we have substitute in getting l 22 equal to 1 and continue with the decomposition process.

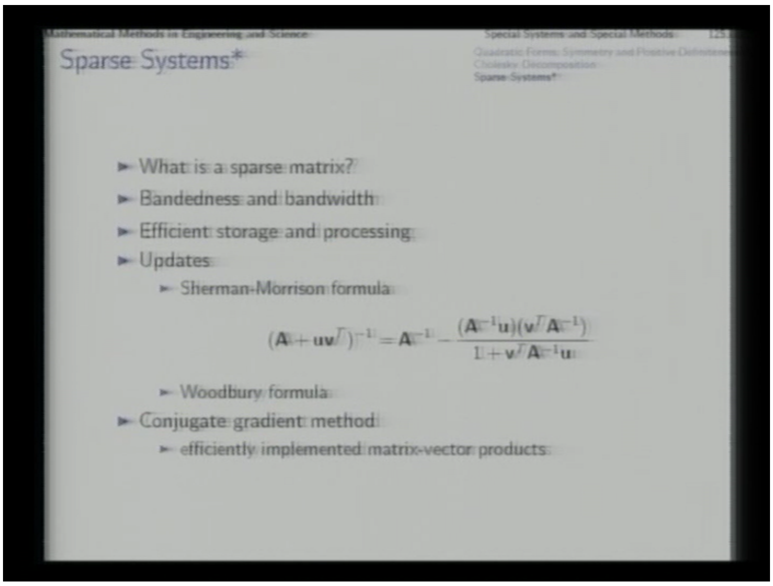
So, from this point onwards I advise you to complete this sequence of operations and see why this positive definiteness which we will get with alpha equal to greater than one is needed for a successful completion of Cholesky decomposition.

(Refer Slide Time: 58:20)



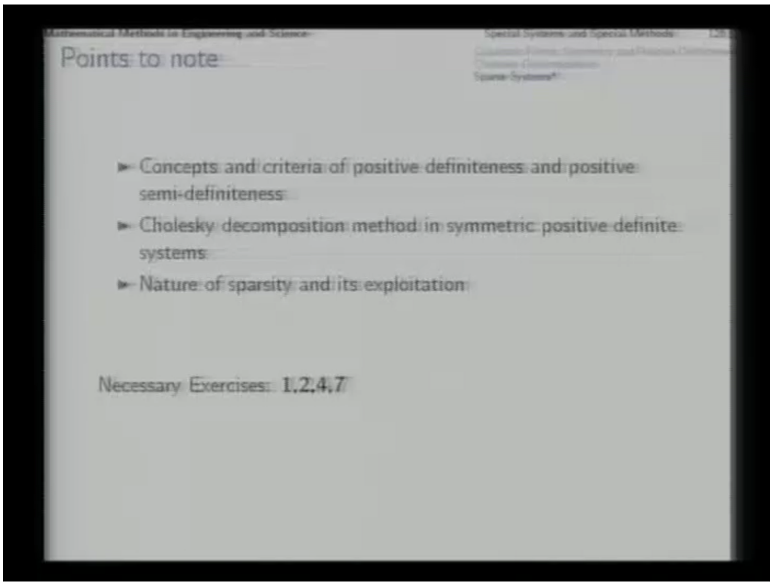
So, this Cholesky decomposition also gives you a test of positive definiteness and it is a very stable algorithm with no pivoting necessary and this says space and time compare to a an ordinary LU decomposition process.

(Refer Slide Time: 58:36)



The rest of the thing regarding the further special structure of matrices is in sparse matrices which we will omit from this course for the time being in the book there is a section on the sparse matrices handling and other particularly advantages of equations which you should go through when you want to covered advance topics in the next lecture we will discuss the situation of particularly bad coefficient matrices which we will handle in the next lecture.

(Refer Slide Time: 59:12)



Thank you.