Finite Element Method & Computational Structural Dynamics
Prof. Manish Shrikhande
Department of Earthquake Engineering
Indian Institute of Technology, Roorkee

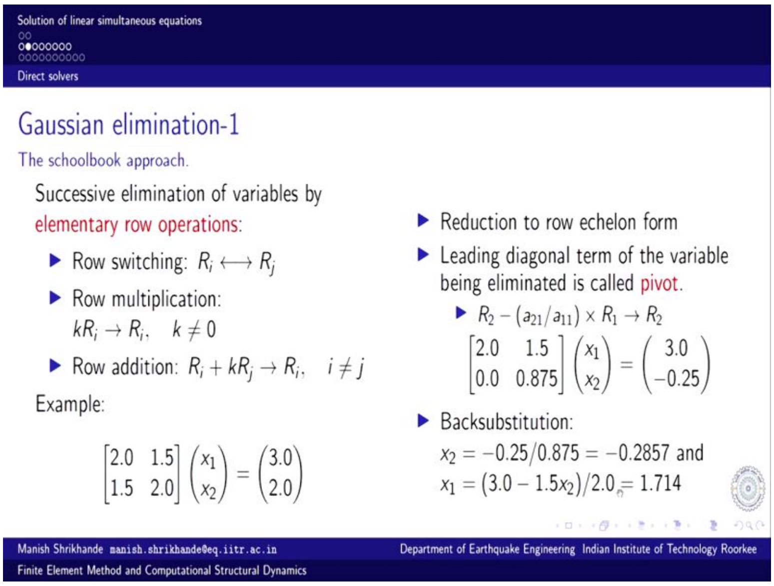
Lecture - 44
Solution of Linear Simultaneous Equations - II

Hello friends. So, we have discussed the basic idea behind Solution of Simultaneous Equations, what is involved, how is it organized that is transformation of the system of equations into an equivalent system of equations which has identical solution as the original, but it is easier to solve.

So, the first exposure to direct solvers that we had in our school was elimination of variables and technically that is referred to as Gaussian elimination. So, we go back to our school days and I call this as the school book approach, how we did while we were at school the Gaussian elimination.

So, essentially it involves successive elimination of variables by elementary row operations that is linear combination of rows does not change the solution, multiplication of row by a constant does not change any change the solution and so on I mean the so, interchange of rows is not an issue.

(Refer Slide Time: 01:49)



And row multiplication by a constant by where k is non 0 constant. So, that is also acceptable, all these operations they do not cause any change in the solution. Row addition I mean two rows can be added and replacing another row that is also possible. So, that is also not going to change the solution.

So, let us take the same system of equations that we had originally so, 2, 1.5, 1.5, 2 this is a symmetric matrix by the way and two variables x_1 x_2; let us call it x_1 x_2 and right hand side vector is 3 and 2. So, the idea is to reduce it to row echelon form by successive elimination. So, that will be leading diagonal term of the variable being eliminated is called the pivot.

So, from this x_1 is being eliminated from equation 2 right. So, when the when we look at the equation 2, it is the equation 2 only involves x_2 and there is no x_1 I mean the coefficient of x_1 is 0 so, x_1 is being eliminated. So, the equation that is x_1 that is being eliminated so, the diagonal term corresponding to x_1 in the coefficient matrix that is this particular element in the diagonal position so this is called pivot element.

And pivot that is important because that goes in the denominator and that is why it is it has special significance we will see why that is so important. So, what is done is row 2 is replaced by a scaled version I mean adding a scaled version of row 1 with row 2 and that resultant is what is replaces row 2. So, when we do this then essentially we scale it down by this vector this scalar pivot element and multiply with whatever was the coefficient here negative time times of that and then add equations at the two equations.

So, that would make x_1 to vanish and resulting coefficients would change I mean this coefficient of x_2 changes and the right hand side vector also changes accordingly. And subsequently this system of equations can be solved x_2 can be solved as minus 0.25 divided by 0.875 and having computed x_2 , x_1 can be solved from the first equation.

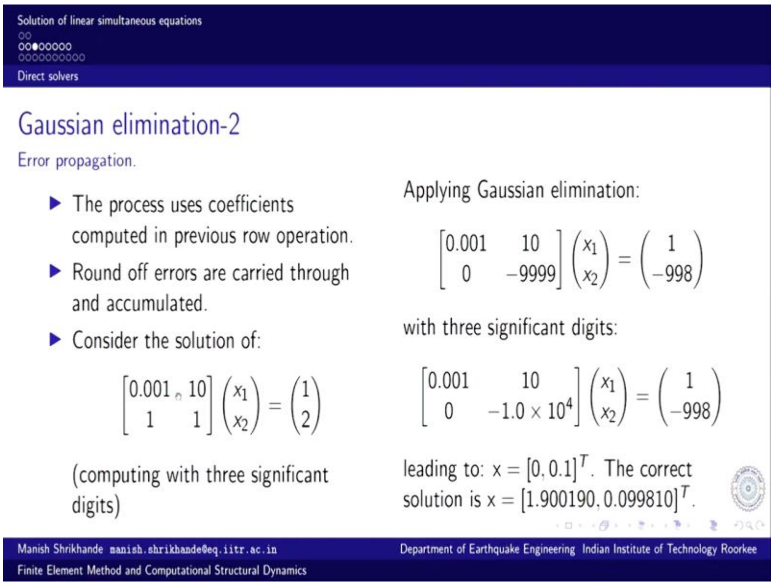
So, back substitution gives us $x_2 = 0.2857$. I am retaining four significant digits here and x_1 is obtained by substituting this value of x_2 in the first equation and x_1 is obtained as 1.714. So, what is wrong with this? This is this seems to have worked very well and these are the standard procedure I mean theoretically we can derive that equations. So, the solution of equations does not change under elementary row operation and once we have this row echelon form the equations can be solved by using back substitution and that is a fairly straight forward procedure.

So, what is wrong here, wrong is in the sense what if this pivot element a_{11} is very small in comparison to a_{21} that will increase the amplitude greatly. And if you recall our floating point representation it is a logarithmic space. So, a large amplitude number incurs larger round off error and that is the whole crux of the problem here in digital computer implementation of this Gaussian elimination.

So, if the system of equations was so arranged that a_{11} the pivot element is for the variable that is being eliminated is very small compared to the other coefficients then the chances are that this elimination process will incur a very high round off error and that will propagate through the solution. Because once an error is introduced that error is carried through the entire process because if I incur an error in computation of x_2 that erroneous value of x_2 will also affect the computation of x_1 here.

So, not only the there is an error in x_2 there is an error in computation of x_1 as well, as a result of the entire algorithm being structured as a successive elimination process. And that is a problem and this is what we call as error propagation, let us look at this.

(Refer Slide Time: 07:44)



So, process uses coefficients computed in previous row operations and round off errors are carried through and accumulated, because one row that is modified is used for subsequent operation subsequent elimination of variables in subsequent operations. So, one round off error leads to another round off error and so on and there will be several round off errors piling on top of each other.

So, just to explain that problem of small pivot let us look at this solution admittedly these numbers are cooked just to drive home the point, that even for I mean 2 by 2 such a simple 2 by 2 system how horribly can things go wrong. So, again I retain three significant digits here. So, 0.00 just to make numbers within reasonable width.

So, again computing with elimination of the variable so, R 2 replaced by linear combination of R 2 and R 1 so, applying Gaussian elimination. So, the solution that we compute is 0 and 0.1 whereas, the correct solution is 1.9 some terms and 0.0998. So, 0.0998 is correctly represented approximately reasonably fine 0.1 is a reasonably good approximation to 0.0998. But this solution this round off value round off result for x 2 leads to completely wrong answer related to x 1 and this is what is a problem with Gaussian elimination Naive Gaussian elimination the school book approach that we have.

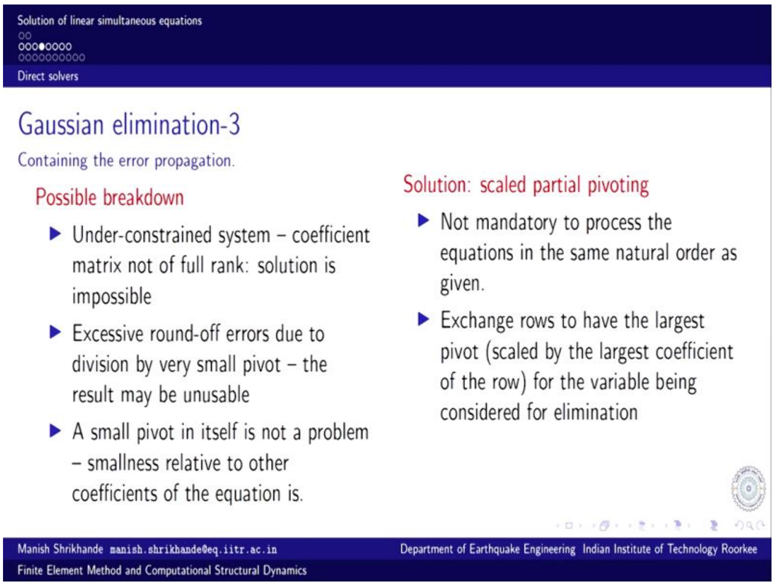
And the entire thing is because of this small pivot element, because once it divides 1 by 0.001 this becomes too large a number I have compared to other numbers and that leads to round off errors and subsequent blowing up of the calculation so, how to guard against this. So, error propagation I mean errors are going to be incurred I mean once we have floating point operation it is a given that every floating point operation will incur round off errors.

So, how do I avoid incurring very large errors or how do I prevent catastrophic propagation of this error, such that the solution is still meaningful, all solutions are approximate any result any number coming out of digital computer is an approximate solution, there is nothing like exact solution exact solution is a myth as far as digital computers are concerned.

So, but the approximation can indeed be a very good approximation. So, how do we get, how do we organize our procedures to get a reasonably good approximation and that is the key is how do we contain the error propagation. So, what is the possible breakdown?

So, possible reasons could be under constraint system coefficient matrix is not of full rank then the solution is; obviously, impossible, but then near singular can also be have we have a problem because the equation that we had earlier I mean the previous problem that pivot element of 0.001 the system is near singular and that is what is leading to problems.

(Refer Slide Time: 13:40)



And that is what leads to excessive round off errors due to division by very small pivot the result may be quite unusable. The small pivot in itself is not a problem, because if all the elements are of relative same order of magnitude if the other term in the equation instead of being 10 had it been 0.002 that would not have been a problem.

So, the smallness is a relative term if the pivot is small compared to other coefficients in the equation then that is a problem, because any scale I mean any row can be scaled by a constant. So, small pivot is not really an issue on its own, but its smallness compared to other coefficients of the equation is a problem. So, how do we deal with this?

So, this is one is scaled partial pivoting what do we mean by that. So, the idea is it is not necessary I mean to process the equations in the same order as they are presented. So, for example, what is listed as row 1 need not be treated as row one I could have handled row 1 as row row 2 as row 1 and row 1 as row 2. If I did that then you would realize that if I interchange the rows the entire problem goes away, very simple the pivot is not very small anymore right we will see how that works.

So, not mandatory to process the equations in the same natural order as given. So, exchange rows to have the largest pivot scaled by the largest coefficient of the row for the variable being considered for elimination. So, we have whatever all rows are scaled by the largest coefficient. So, we have normalized equations all row all equations are normalized to have largest coefficient being one in each row.

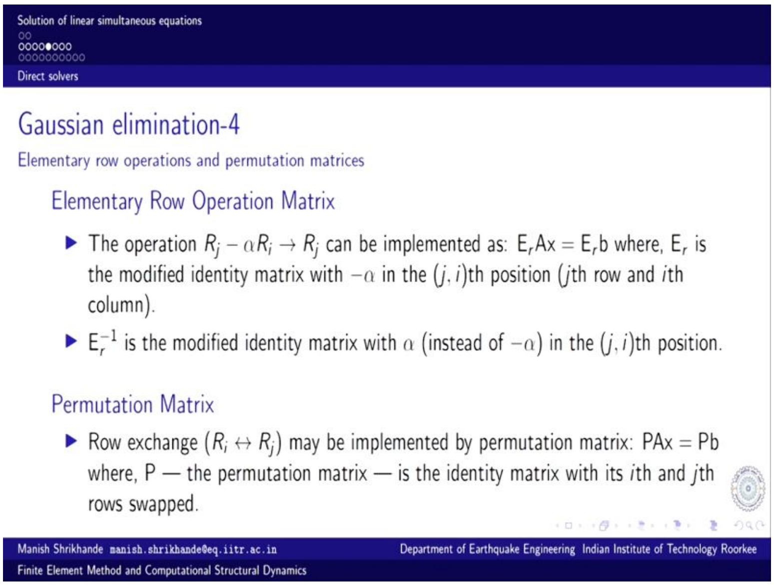
And then we look at that particular row for considering pivot as which has the largest coefficient and then we implement exchange of rows. So, that would eliminate the problem of using a very small pivot which will lead to blowing up or amplification of round off errors.

But this pivoting row exchanges destroys LU structure of Gaussian elimination for example, if we store the multiplication the whatever coefficients that we have for implementing that Gaussian elimination for example, a 2 1 by a 1 1 minus of a 2 1 by a 1 1 in the position a 2 1 because that coefficient would anyway be 0. So, we can store this coefficient.

And then this off diagonal triangular structure and keep diagonal terms as 1. So, this becomes the lower factor lower triangular L and the other one is of course, U. So, this becomes LU factorization of matrix A and this is only possible when I do it in the once I decide the order and I process the entire system in the maintaining the same order, no further row exchanges anywhere, but if I do this row exchanges pivoting exercise then it prevents it destroys this LU structure. So, that is one thing one minor result that might be of interest.

So, essentially what is done is it is again row interchange I mean mathematically they can be implemented very easily and there are row operations implemented as a scalar multiple and permutations interchanges can be done by using permutation matrices if required for the analysis purposes.

(Refer Slide Time: 18:39)



And then eventually they can be implemented as elementary row operations of matrix A same elementary row operations are performed on matrix b. So, E r is the modified identity matrix. So, we can interchange the identity matrix interchange the rows and that becomes the permutation that will once we multiply with this. So, it will permute inter exchange the rows interchange the rows.

And similarly this E r is inverse is simply computed by substituting α instead of - α in the i^{th} position. So, row exchange can be implemented by permutation matrix and that is just the interchange of the identity matrix and this elementary row operation that is done by using a scaled version of I mean linear combination of two rows. So, that is done by replacing the identity matrix coefficient to j^{th} i^{th} position as - α and then multiplying that.

So, this is how it can be implemented very easily on the it is very easy to develop a code on the basis of this permutation elementary row operation matrix and permutation matrix and then the sequence of operations can be defined. So, the entire system entire Gaussian elimination can be carried out as a series of matrix multiplication, elementary row operation, matrix permutation matrix, elementary row operation, matrix permutation matrix and so on.

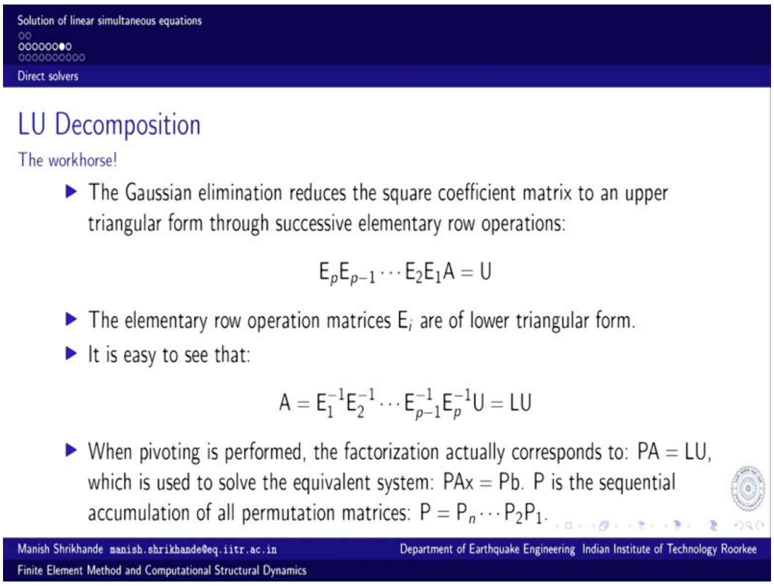
The problem with this approach is there is a large scale movement of data in computer storage and that data is stored in a slow speed storage area and this will slow down the operation execution of the code. And that is what I refer to as steep performance penalty.

So, what is essentially practical implementations are done is by making little bit of a book keeping track I mean we do not interchange any data. So, the purpose of we realize that the purpose of partial pivoting is merely to establish a preferred ordering of equations for sequential elimination of variables. So, as long as I develop a mechanism to identify which equation which variable has to be eliminated from which equations in what order, then this interchange of rows etcetera is not really required it can be done in place.

And that is what is done I mean by keeping track by using an index array integer index array and that is a it keeps track of which row is to be considered as a pivot row for elimination of which variable and using that it can be the entire Gaussian elimination process can be computed and this index array position can be used for back calculation or back substitution operation for calculation of the solution and the while computing the variables solution for variables.

So, this is all for Gaussian elimination then we have other approach that is LU decomposition matrix A can be decomposed as a product of lower triangular and upper triangular matrix, again as I said Gaussian elimination process itself is a if I just use elementary row operations.

(Refer Slide Time: 23:04)



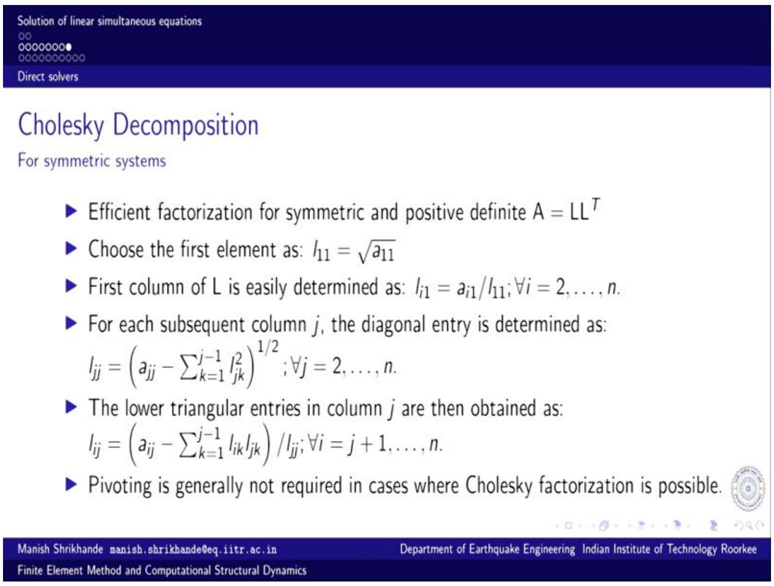
That itself is a it will convert it into upper triangular matrix and this entire process is a lower triangular matrix. And its inverse operation is easily computed by just changing the sign of that multiplier alpha instead of minus alpha it will be plus alpha in the inverse of the matrix, otherwise E matrix are essentially identity matrix except for jth ith element.

So, it is easy to see that matrix A is simply inverse of this. So, while we take the inverse it will be order will be reversed. So, that is essentially L times U. So, that is the LU decomposition and once we have this LU decomposition these equations can be solved by two step process. So, when pivoting is performed this factorization actually gets affected because this will not be related to a after pivoting, but rather the permuted form of matrix A PA.

So, there are ways of handling that, but more I mean this is theoretically it is possible that Gaussian elimination can also lead to LU factorization. But it is more prudent to use symmetric positive definite matrix can be decomposed using what we know as Cholesky decomposition algorithm and in that case upper triangular matrix is going to be transpose of the lower triangular matrix. So, U is simply transpose of L and matrix A is given as LL transposed.

And once we implement this lower triangular matrix and implement the product we will find that we can actually work out one to one correspondence between the unknown equations unknown terms of mat lower triangular matrix L and the coefficients of matrix A.

(Refer Slide Time: 25:25)



So, L_{11} is computed as square root of a_{11} and then rest of the elements of first column can be computed by simply taking ratio of a_{i1} divided by L_{11} and similarly then subsequent column j it can be similarly arranged by this standard algorithm.

And once we have this the lower triangular entries for all the columns are available and once we have this Cholesky factorization then subsequently the solution of simultaneous equations can be computed by using two stage process forward substitution and backward substitution and then the solution complete solution is known.

Wherever Cholesky factorization is admissible the system of equations is generally well conditioned and no further pivoting is generally required of course, it is always a good practice to first estimate the condition number of the matrix, just to know whether the system of equations is well-conditioned or not.

So, that completes our discussion of direct solvers, you may explore this topic further and work out some of the try to develop code based on the basis of this Cholesky decomposition and also Gaussian elimination algorithm that has been discussed and try to implement the partial pivoting by using the index array that would be a great learning exercise.

In our next lecture we start with iterative solvers and that is a very powerful set of techniques and of immense application.

Thank you.