# Matrix Method of Structural Analysis

**Prof. Amit Shaw**
**Department of Civil Engineering**
**Indian Institute of Technology, Kharagpur**

## Lecture - 31
## Computer Implementation

Hello everyone, this is the 7th week we are going to start today. You see we are almost towards the end of this course. If we just look back at this stage, in the first 2 weeks we discussed we briefly reviewed the structural analysis 1; then 3rd week where a brief review of outline of the matrix methods matrix algebra. And then 4th, 5th, and 6th week in last 3 weeks we discussed what is matrix method of structural analysis, and how it is to be done for different structures. We started with truss and then we extended the similar concept for beams and then frames.

And of course all the methods that we discussed in the case of cross beams or frames those the various steps involved in the analysis were demonstrated through some examples. You see if you recall, one of that we discussed in the very beginning of this course to the very first day of this course, any method you use one of the important criteria that method should have is the adaptability and scalability.

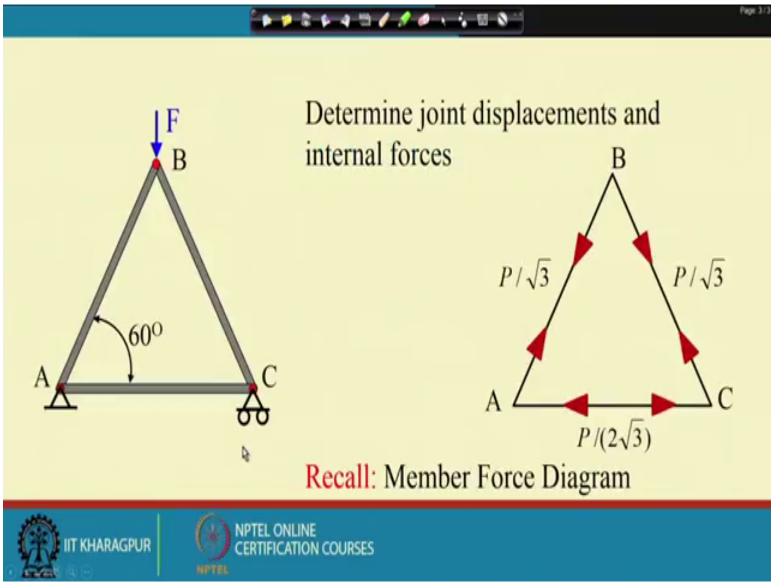
You see the motivation behind starting this method moving beyond structural analysis 1 is the methods that we studied in structural analysis 1, those methods are not scalable. Scalable in the sense for small problems we can do those methods require tedious manual calculations, which is not feasible, when structures are very large right, when you have several numbers of beams several numbers of members and nodes.

And one of the important motivation for studying or moving beyond that method is to look to find a method through which we can solve large of the problems with very large dimensions with large dimensions. And the large dimension means in this case very large number of degrees of freedom. And whenever we have a such a problem with large number of degree degrees of freedom, manual calculations cannot be done.

In the last three weeks all the demonstration for the demonstration purpose in order to understand the various steps in a detailed way, we did all calculations manually. But now when we have a structure which is having many number of members and many number of nodes that manual calculation not possible. And so we have to implement them through computer code right. And the method the method that we have been learning in this course, the one of the important feature of the method is it can be translated easily, it can be translated into computer codes.

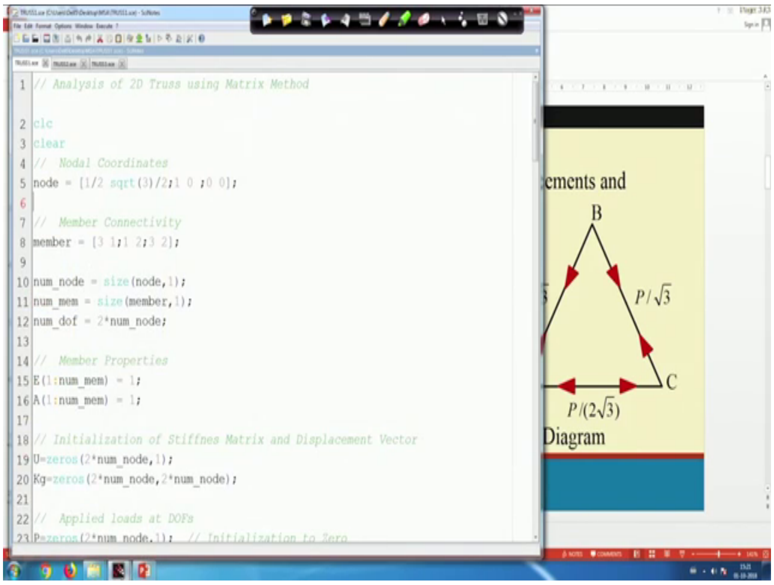
What we will discuss this week is, we will see some of these implementations, whatever steps we discussed in last three weeks, the similar steps will translate into computer code. We will see how those computer programs to be written. You see whenever we talk about computer program, this is very subjective. Subjective in the sense the naming different variables, the way the loops are written the way, the variables are stored, the array structure all things are very subjective. So, here idea is not to discuss that; here idea is to discuss different steps and how crudely how those steps can be performed. And then we will take a very large problem and then see with a just mouth is a mouse click we can have the solution of those problems ok.

(Refer Slide Time: 03:50)




So, to start with let us say this example, because these example we solved here in the class, we know the solution. Let me demonstrate these steps the computer code through this example ok. Now, this code is written in Scilab this is a open source software. You can download it.

(Refer Slide Time: 04:16)




So, these codes will be uploaded on the forum, but here the our purpose let us understand the code here, and how to use this code. And then of course, you can modify this code, you can write this code, you can make this code more efficient you can change the number in you can change the variable names the way you want to modify the code ok. You see now consider this, if you recall, this is an important this is called discretization right.

Now, we have 3 members here member number 1, 2 and 3. And this is how the member numbers are written, and the corresponding degrees are degrees of freedom are identified. And this is member number and this is corresponding i and j; i and j is very important, because here for this member if you recall for this member we say that this if this member is 3 1 means, we are moving in this direction means ith will be ith node will be 3 and jth node only bond this we discuss.

Similarly, lambda x lambda y can be obtained like this once we have this then the element stiffening the member stiffness matrix for a given member the general form of the member stiffness matrix is this right and these four column, because these are 4 by 4 matrix because for a given member we have only 4 degrees of freedom ok. So, whenever we have a problem we create the model and then a first thing we have to do is we have to identify, we have to give the coordinates of the nodes ok. For instance, in this problem we have 3 nodes, so these are the coordinates of the nodes. So, this node is 00, this node is L 0, this node is this.

So, in the computer code, let us say clear and clc, it is to clear this screen and also to rearrange the memory of the previous variables. So, once you when say clear, it does not store any variable, all the memories clear. Now, node is in this case the node is the coordinates of each node, so we have 3 nodes. So, first node is node number 1, made node number 1 is this node and whose coordinate is L with L by 2 root 3 L by 2 coordinate number, coordinate of node 2 is 1 0 or L 0, in this case it is 00. So, node number 1 node is this is the x coordinate of node, and this is the y coordinate, we assumed the length L is equal to 1. And this is the second node, and this is a third node.

Now, for a if we have a different problems, then for different problems, once we identify once we give the node numbering, you see that how to give node numbering is an important part, but let us not bother about right now, because we give some way we numbered them, get the global stiffness matrix first understand the code. Then we can see for different numbering system what happens to the global stiffness matrix, we will see that.

Now, so once we have these node numbers, now next thing we have to give these information, how these elements, how the members are connected between nodes. Now, when we say how the members are connected between nodes, we have to give what is the ith node of the member, and what is the jth node of the member. So, this essentially these is, so member number 1 connecting between member number 3 and joint number 3 and joint number 1; member number 2 is between joint number 1 and 2; and member number 3 is between joint number 3 and 2 ok. So, this is the information that we have to get from the configuration of the structure right.

Now, these three is once we have this, you can get this information from suppose for a complex structure, you it is very difficult to write the nodes and member connectivity like this. What you can do is, you can use any software to create the model. And once you create the model, export the information about the node and the member connectivity, and use that information in this code.

So, these automatically gives you, what is the total number of node we have, and this is the total number of member we have. Size is a command which gives you the size of a matrix or a vector. And num number degrees of freedom is what is the total degrees of freedom in the system. You see per node it is a truss per node we have 2 degrees of freedom. So, total degrees of freedom will be 2 into number of node ok.

Then comes the properties of the member, properties of the member means what is the if you look at the properties are required the Young's modulus, which is the material property and then cross section of this material cross section of the member. Length it automatically calculates from the joint coordinate. So, here we have to give what is the property of this member.

Now, let us assume you can give any value. So, we have given just for demonstration say a is equal to 1 ok. Now, these properties can be different for different material in this case it is given as E is equal to 1, and A is equal to 1 for all the members. But if you want if you have a structure which having different cross sectional areas, different material, then you can change it. We can this is essentially a vector the size of the vector is the same as the number of member.

Similarly, size of this vector is a number of members. So, we can change the values in that vector. So, this is the member properties. Now, once the member properties are determined, now almost we have all the information about the truss geometry of the truss right. Now, we will come to the load vector later, but in order to construct the stiffness matrix, what are whatever information we need then information right now is provided.

Now, before we calculate the stiffness matrix, it is a very important part to initialize the variables. You know sometimes even if you do not initialize the variable, if there is a bug in the code, then those you can those variables may get some garbage value. So, it is better always to initialize the variable variables. So, Kg is the global stiffness matrix, size of the global stiffness matrix will be total degrees of freedom by total degrees of freedom. And then this is the load vector; size of the load vector will be the length of the load vector will be again the total degrees of freedom ok.

Now, this is the initialization of stiffness matrix and displacement vector. Now, next is applied loads at degrees of freedom. Now, you look at this problem, here the problems degrees of freedoms are like this. If you degrees of freedoms were like this, this is degrees of freedom 1, this is 2, and this here it is 3, and it is 4, and then here we have 5 and then we have 6. So, what are the forces we have? We have forces F here in this direction, so along degrees of freedom 2. So, if P is a vector then we have all quantities at 0, and it will be minus F and 0 0 so on ok.

So, similarly we have now this is P is equal to 0, it is initialization of the entire load vector, and in that load vectors a P 2 is equal to minus 1. And the force correspond to the second degrees of freedom the minus 2 can change the value, if you want. Now, once this is done then we have to give the boundary condition. So, what boundary conditions we have here is? The boundary condition we have is this. You see this is a hinge support, and this is a roller support. So, 5 is degrees of freedom 5 is constraint, 6 is constraint, and degrees of freedom 4 is constraint. So, we have constrained A along the agrees of freedom 4, 5 and 6. And then the degrees of freedom which do not have a curve which does not have constraint is 3, 1 and 2.

So, similarly so we can provide this value here U known, U known means, where you have the constraint means U is 0 there, so in this case. So, U known is 4th degrees of freedom 5th degrees of freedom, and 6th degrees of freedom. And U unknown means the rest of the other degrees of freedom, where there was no constraint those are the displacement that we need to determine from this from the analysis. Now, this is 1, 2 and 3, now this is the boundary conditions.

Now, next let us if you recall when it we will come to this point later ok. Let us find out the now stiffness matrix once stiffness matrix and then assemble them. This is this is an important part. And you see when you write do not take this, as I say this code will be uploaded in the forum, but do not take this code (Refer Time: 13:52). It is always better in order to understand the steps involved. In a better way it is always desirable for you to write your own code ok.

Now, this step is important. This step is calculation of element stiffness matrix or the member stiffness matrix, and put them in the global stiffness, and the assemble them in the global stiffness matrix. If you recall how we deduct first, we calculated the element stiffness matrix; here we have three members, so we have three members stiffness matrices. And the global stiffness matrix is in this case will be 6 by 6, because 3 nodes. So, first we initialized if you recall, initialize the global stiffness matrix, all the elements in the global stiffness matrix is 0. And then we took the we then we check these element stiffness matrix degrees of freedom by degrees of freedom element wise and then that is added to the global stiffness matrix.

So, you can again revisit the lectures the previous lectures to understand the exact steps ok. So, this loop will be over the number of members, so num member is the number of members. Then this is the length calculation of each member. And then once we have length calculation then if you recall this is the member stiffness matrix, where we need lambda x and lambda y, lambda x and lambda y essentially the measure of theta. So, this gives you lambda x, c is the lambda x, and s is the lambda y ok. Now, once we have calculated lambda x, and lambda y.

The next thing is this is the elements in member stiffness matrix. Now, the member stiffness matrix is, this is the member stiffness matrix Ke is the member stiffness matrix. This member stiffness matrix it depends on what is the value of e, because it is sum over e, now this is the element member stiffness matrix for a given e. Now, what is then, we have to we have to assemble them right. Now, you see already we have initialized this member stiffness matrix 0 0.

Now, once we have then what is the process? Process is to we check the element stiffness member stiffness matrix, and take the corresponding quantity corresponding elements in the matrix, and put and add to a global stiffness matrix. And this is done in this case well this is done here, this is the thing this is done in this part ok.

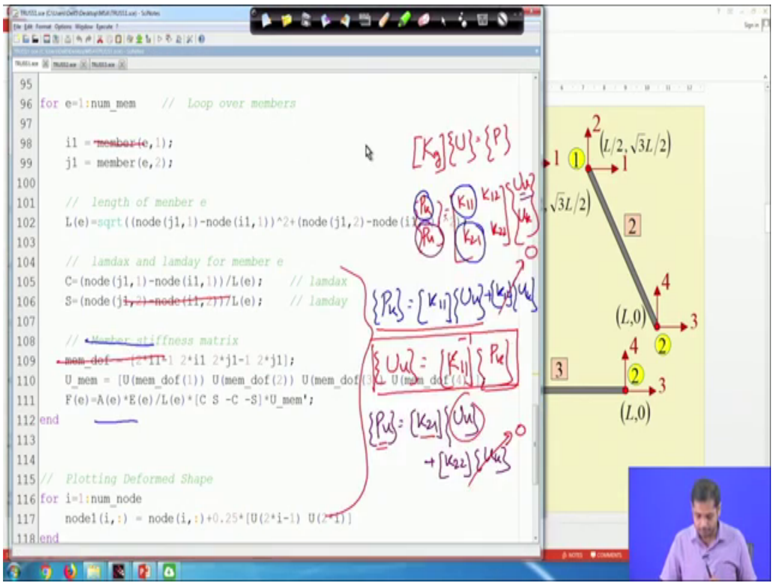
So, this gives you if you recall, when we write when you wrote you see when we have the element member stiffness matrix for a given member it is i 1, i this is for these this is for i and j, but i and j are different for different members. So, therefore, what will be in the global with respect to the global coordinate system, what will be the i 1 the index indices of i 1 i 2, j 1 j 2 that depends on the connectivity, and the degrees of freedom.

So, similarly we can. So, this is the global assembling of the stiffness matrix. Now, this entire part has to be done over the entire element. So, first e takes value 1 for a member number 1, calculate the member stiffness matrix. And then that member stiffness matrix is substituted in the corresponding position in the global stiffness matrix.

Then goes to element number 2, member number 2, then member number 2 stiffness matrix calculated. And then similarly in the global stiffness matrix which is all, which has already been populated for member number 1. And then it has to be populate from member number 2 as well, similarly when A is equal to 3 it has 2. Calculate the member stiffness matrix for member 3 and then whatever populated matrix we have after member 2 that is again amended for member 3. So, this is how the members take global stiffness matrix is being calculated ok.

Now, once we have the global stiffness matrix if you recall, with global stiffness matrix is if you try to find out the determinant of global stiffness matrix, the determinant of global stiffness matrix will be 0. And the physical manifestation of this is we have not provided any support to the structure therefore, structure is unstable. Now, therefore in order to main this structure stable, we have to provide this support providing support means that information needs to be given to this stiffness matrix and that is done and that exercise. If we recall, that exercise is called partitioning of the stiffness matrix.

(Refer Slide Time: 19:11)




What is the partitioning of the stiffness matrix, it was something like this. So, if it is global stiffness matrix it is K, if you recall if global stiffness matrix isK, it is K g here anyway. So, this is K g and then this is U, the global force like of displacement vector. And then we have global force vector. Now, these can be written as if you recall, then that K 11, K 1 2 K 2 1 K 2 2. And then we have this is U unknown, U 1 known, and U known, and this becomes P known P unknown, if you recall this is how we did the partitioning.

Now, U unknown and U known is already known. Already we have given this information. What is the U unknown, what are the degrees of freedom are known, and what are the degrees of freedom known, that is with that information we have here. Total force vector is also there only depending on these indices in P unknown, U unknown and U known we have to restructure you have to reorient the corresponding members in the global in the force vector.

And similarly the elements say this has to be done for the global stiffness matrix as well. And this is done here, this is done yes partitioning of the stiffness matrix is done here. Now, not only the partitioning of the stiffness matrix you have to correspondingly partition the load vector as well. So, this exercise this entire loop what it calculates is this. These entire loops these entire loops these entire loops calculated calculates K 11, this is K 11, which is k 11 here and then K 2 1, K 2 1 here ok. Now, and then P known is this is calculating ok. P unknown is not required, now these three things calculated in this loop.

Now, once this is calculated, and if you recall, what we did is, then from this we can write P k is equal to K 1 1 into U unknown plus K 1 2 into U known. Now, U known is 0 here anyway, U known is 0, U known is 0, so this part is anyway 0. So, essentially we are left with this part. And from this we need to find out we need to solve this linear system or equation for U unknown. U unknown is equal to K 11 inverse and then P k, but I am just writing inverse for the representation. If you recall, it was discussed in the 3rd week that if you have a linear system of equation, then we do not need to invert the coefficient matrix, because inverse is a very computationally intensive exercise.

Instead what we can do you can solve you can do L U decomposition and then solve it there are methods to do that. It is just not directly inverting the stiffness matrix and get the solution ok. So, you know how to solve the linear system this is exactly done here to get U unknown. Now, this exercise is done next yes. This is solution for unknown displacement. Lin solve is the command which actually solve it this is a command for scilab. So, this you get U u.

Now, once you get U u; U u is known in this expression if you take different colour, now U u is known , then come to we have to find out what is P U, P unknown. P unknown gives you essentially the, this P unknown is essentially the forces where the constraint where you have the constraint. So, this essentially gives the support reactions ok. Now, in order to get the support reaction we get the second set of equation which is p unknown. If you recall, that is equal to K 2 1 into u known U unknown. U unknown we already, now we have already determined the value of U unknown and then plus K 2 2, K 2 2 into U known ok.

Now, since again U known is 0 here this part is 0 that is why this K 2 2 is not calculated here. So, K 2 1 is already calculated here. This is K 2 1 we know K 2 1. U unknown already solved for U unknown so this solution is already with us. And then from this expression we get P U u. So, this gives you the support reactions ok. So, once we have the support reaction next part is find out the member forces ok.

Now, if you recall how to get the member forces, this exercise is done here. This is member forces. Now, if you recall the member forces was if we just yes. Now if you recall the member forces was for any member, if you want to calculate the member forces, that member force will P. If we take say P dash, the member force in a given member that is equal to, if you recall AE by L into lambda x, then lambda y minus lambda x minus lambda y and then your U; U for this member. So, this is U for this member means, this member has 4 degrees of freedom and that corresponding to that degrees of freedom what is the component of U here. So, this exercise is done here. So, this is again loop over the over all the members, this is the entire loop, this is the entire loop over all the member, and you get the member forces here right.

Now, once we have the member forces then next is then you can have do all the information you can plot them and do rest of the things. So, you see demonstration of this code will discuss in the next class through some example. We start with this example, which is a very smaller one, which we have already solved in one of the classes. Then we also take another example, which is relatively large; relatively larger in terms of number of members, and the number of nodes relatively larger. And then we also try to solve them using this and you can take this solution from this code you can try this and then verify the solution ok.

So, this is the code, this code is a you know when you actually see, actually this is a very code for written for academic purpose, just to demonstration the just to demonstrate the implementation of the methods that we discussed in last three weeks ok. Now, you can write your own code, you can have a better definition of variable, better structuring, better storage of variable, which code is very scalable, but the pre essence of this different steps is this ok. So, I stop here today see you in the next class.

Thank you.