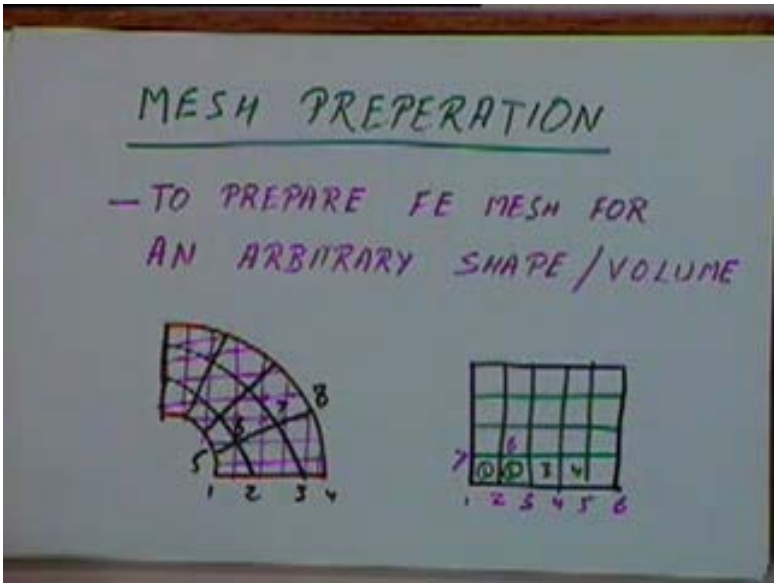
Computer Aided Design
Prof. Dr. Anoop Chawla
Department of Mechanical Engineering
Indian Institute of Technology, Delhi
Lecture No. # 29
Mesh Preparation

Today we will see how to prepare a mesh for any arbitrary shape or any arbitrary volume in the case of sedimentary objects. The basic problem is that if you have any arbitrary shape like this and we want to make a finite element mesh in it. We cannot make any arbitrary mesh because let's say if you want to make a mesh which is just like this consisting of rectangular elements.

(Refer Slide Time: 00:01:09 min)

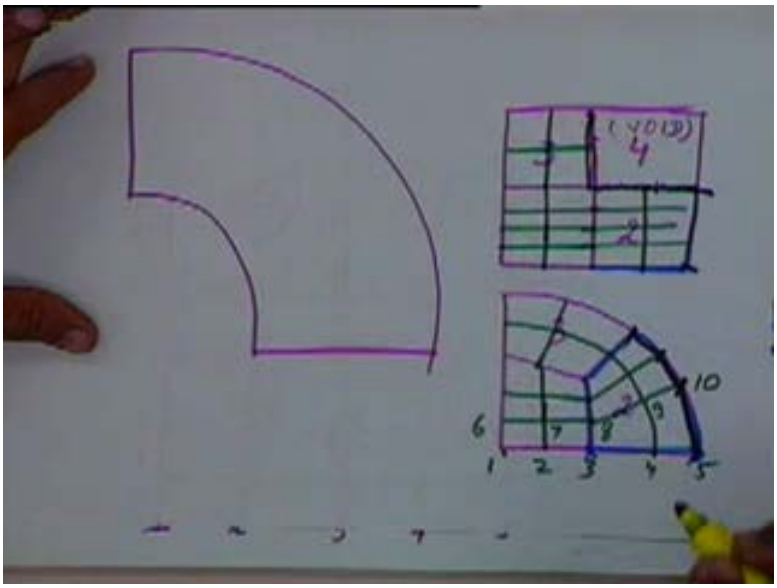


Firstly you might not get the best result in this case and secondly the numbering pattern might also not be prefect and our boundaries might also not come out to be very nice. The biggest problem would be that when we want to make a mesh like this, making it manually is not going to be possible because we will have to get the geometric corners of each and every node. So making a mesh for any arbitrary shape, we have to have some automatic methods for doing that.

For constraints of this, I might like to make a mesh which would look something like this. That means each of these is a quadrilateral element, a 4 nodded or a 8 noded quadrilateral element and we get a mesh like this with numbering going 1 2 3 4 5 6 7 8 and so on. This might be one possible way in which I might like to measure but I need some automatic method for generating this mesh. If you just have a simple rectangular object that means it is an object like this. Then generating a mesh is a straight forward thing. All that I have to do is take a rectangular array of blocks. If I take a rectangular array, I will get an array something like this. So for this array I can generate my node numbers and so on. Let's

say this is element number 1 2 3 4 and so on and my node numbers can be 1 2 3 4 5 6 and so on, 7, this one will be 8 etcetera. But if I have an arbitrary shape and getting the coordinates of each and each and every node, we need some algorithmic method for doing that.

(Refer Slide Time: 00:03:56 min)



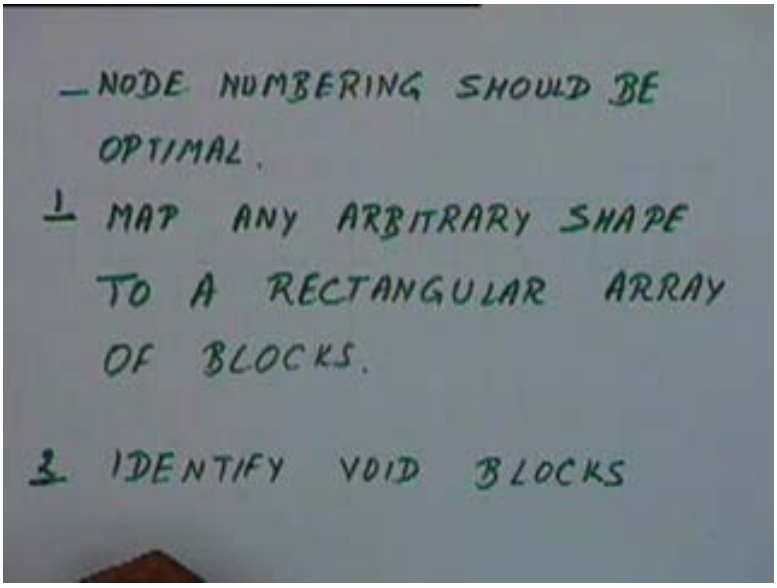
So what we do is if you have any shape, let's say the shape that I have drawn here. We will try to map this shape onto a rectangular block that means if I have a rectangular block like this, I will try to map this shape onto this block and then in some sense I like to twist the elements so that the two shapes become identical. I just explained what I mean by that. For example if I take this block divided into 4 and then I say this is block number 1, block 2, block 3 and block 4. Now I say that this block 4 does not exist that means this block is a void block and then I say in addition to that this edge and this edge, these 2 edges are merged together.

So I will combine these 2 edges so that this node will identical to this. Any point here will become identical to a point here and so on. Then this shape would become something like this, so this these this edge and this edge have been combined together. So what I will get will be some shape like this. This will be block number 1, this is block number 2 and this is block number 3 and the block number 4 which did not exist has vanished. Now this is a rectangular array of blocks. I can do my node numbering in this and that can be mapped onto this shape.

Similarly for this, we will take a rectangular array of blocks and map that rectangular array onto this shape. The basic idea is that let's say in each of these blocks, I decide that we will have a mesh. Let's say in this portion I have a mesh of this size, here maybe I have a mesh which is finer, here also maybe I have a finer mesh and here I have a mesh of this size. When I join them, what I will get would be in this region my mesh is going to be close. So here I will get a mesh which will look something like this. Here I will get a

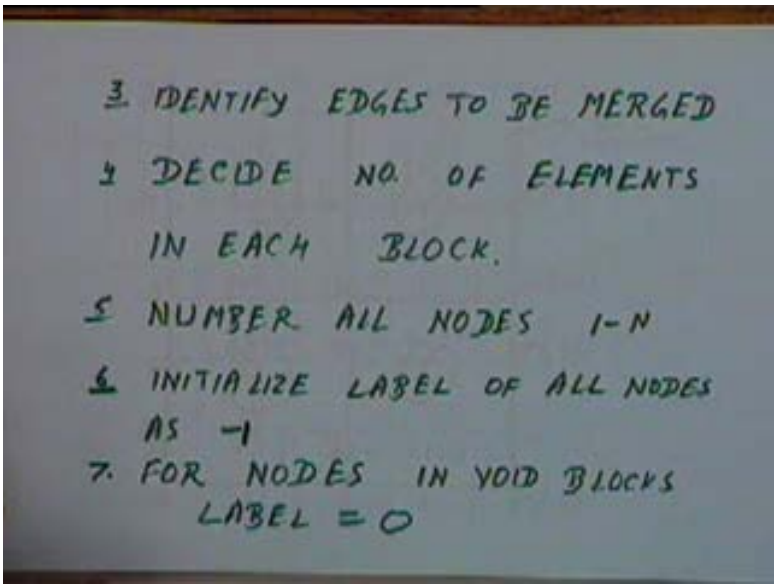
finer mesh and similarly here also I will get a finer mesh. So depending on the portions where I want to have a fine mesh, I can choose a different mesh density in different blocks and then merge the edges which are to be joined together. So for this complete process, we have a simple algorithm which will which after this mapping has been done that means after an array of blocks has been mapped onto a given shape, we can generate the element numbers and the node numbers. But we have to be careful about one thing that the node numbering should be optimal or should be or should give us good results because we have already seen that if our node numbering is not good that means in this let's say 1 2 3 4 5 6, if you don't have a appropriate node numbering then our stiffness matrix is not going to be a banded matrix. So we will not get a good stiffness matrix.

(Refer Slide Time: 00:08:06 min)



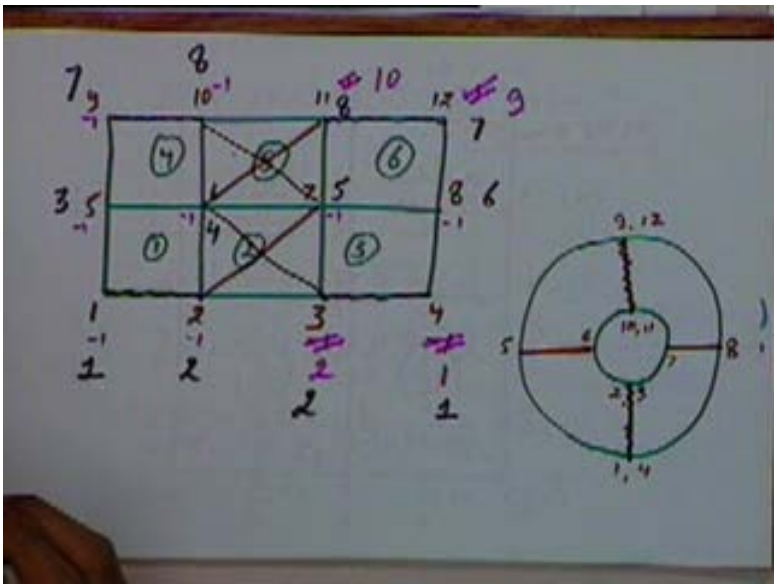
So therefore we have to be careful about the node numbering in these algorithms. That is a critical thing as far as any of these mesh preparation algorithms are concerned. And in this algorithm, the first step is that we have to map any shape or map any arbitrary shape to a rectangular array of blocks. So you have to map any arbitrary shape into a rectangular array of block that is the first step that we have to do in any algorithm. Then this is let's say the first step, after this you have to identify void blocks like in this case we said that this block 4 is going to be a void block.

(Refer Slide Time: 00:10:51 min)



And once you have identified the void block, the next step is to identify edges to be merged. That means in this case we said this edge is to be merged with this edge. So this way we might like to merge different edges in this array of blocks. After we have done this then we have an array which should look something like this and we can have different types of mappings.

(Refer Slide Time: 00:11:43 min)

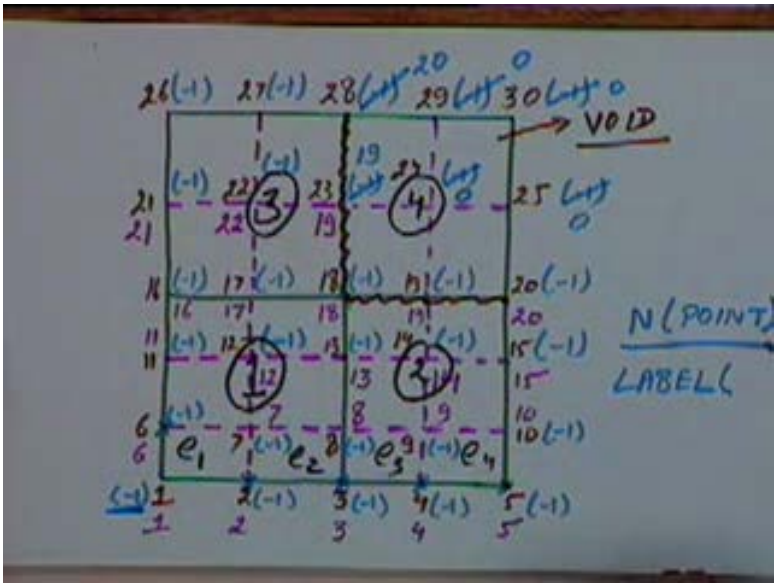


For instance if we consider this array of blocks, this is block 1 2 3 4 5 and 6 and our final object is let's say a circular plate with a hole in it. Then what we can say is that we will take this edge and this edge and we will merge them together. If I just give them numbers

1 2, I will take this edge and this edge I will merge them together and I will consider these two blocks to be void. So if I consider block 2 and block 5 to be void and I consider 9 10 to be merged with 12 11, 9 10 is to be merged with 12 11 not with 11 12 that means 10 and 11 will be merged together and 9 and 12 will be merged together. Similarly 2 and 3 will be merged together and 1 and 4 will be merged together. Then at this shape would be divided into 4 blocks which should look something like this. So this would become 6, this would be 5, this will be 9, this is 10 and the same time this will be 12 and 11, this is 7 8 and this is going to be 2 and 3 and this is going to be 1 and 4. This is one merged edge and this is also a merged edge.

So this, in this manner with a whole can be mapped onto this array of blocks and then we will get a mesh for this array keeping in mind these edges are merged and these blocks are void. This way if you have any arbitrary shape, our task of preparing the mesh will finally come down to the main task that we have to mesh this arbitrary shape into an rectangular array of blocks like this. So, after this stage after we have identified the edges to be merged, the next step is to decide number of elements in each block, how to decide the mesh density.

(Refer Slide Time: 00:15:50 min)



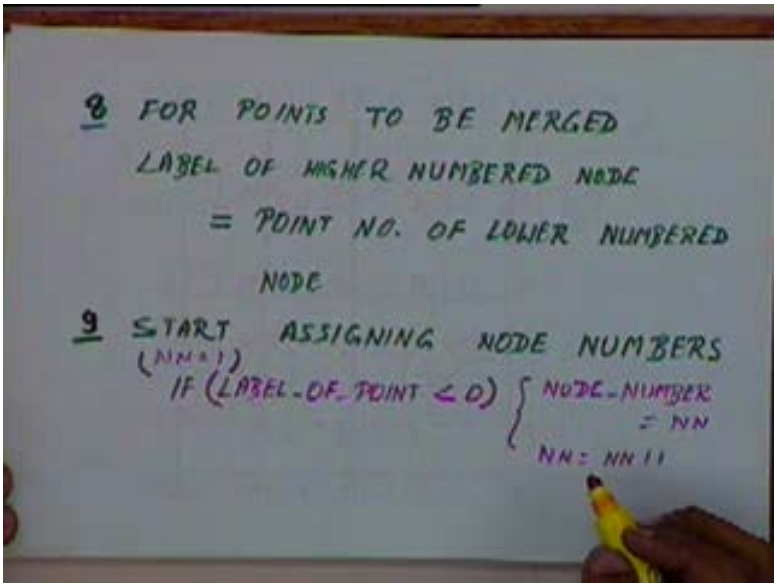
So let's take the simple example that we had earlier. This is let's say for this particular case that means this quarter of a circle or quarter of a circular plate is being mapped onto the array of 4 blocks. So we get these 4 blocks, let's say this is number 1 2 3 and this is 4. Then maybe we decide that we will have a mesh density of 2 by 2 here and a higher density here and out of this of course block 4 is a void block and this edge is being merged with this edge. This is a void block. So we will carry out the complete mesh generation algorithm on this array of blocks.

So this step was to decide the number of elements in each block. The next step is number all nodes, let's say from 1 to N. So in this case the numbering can be done which should

look let's say this is 1 2 3 4 5 6 7. We have to number all the nodes including the nodes inside the void block right now. So this one will become 24 and in this case we number them from 1 to 30. Then for each node we define a label, for each of these nodes we will give a label to it and we will say that initialize the label of all nodes as minus 1. So for each of these nodes, we will give it a label and that label let's say minus 1 for each of them. So let's say this is, so all these node will initially be given a label of minus 1 and then we will go and change these labels. So that's the initialization step we are carrying out.

Then for nodes in void blocks, we will put the label equal to 0. For nodes which are inside the void block, we will put the label equal to 0. That means that this label will be changed to 0, this will be changed to 0, this will be changed to 0 and this will be changed to 0. These labels which are on the boundary they will not be made 0, only those labels which are inside the void block. These labels are common to void and a non-void block. So these 4 nodes, their label will be made 0. The basic idea of making them 0 is that in the final numbering, we don't want these nodes to be included. So we will make these 4 nodes, the label of them to be 0.

(Refer Slide Time: 00:22:13 min)



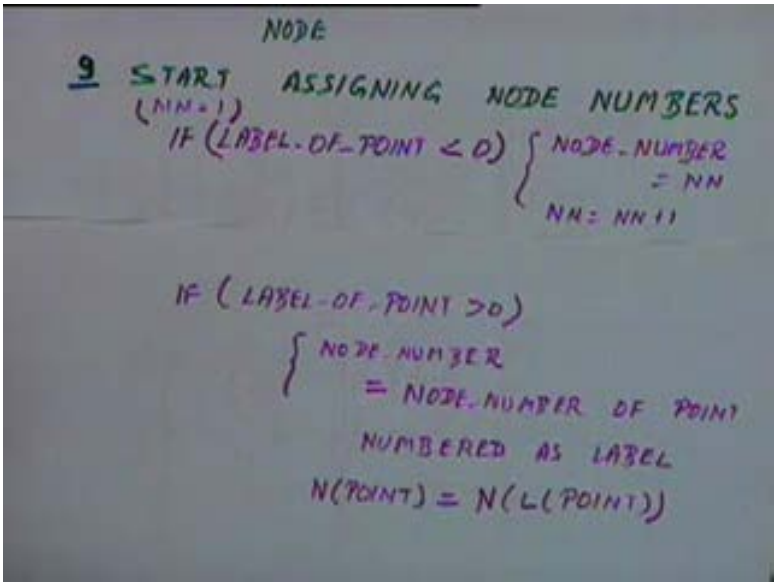
Then we say the next step is for points to be merged that in this example node 28 is to be merged with node 20, node 23 is to be merged with 19 and 18 is of course common to the 2 edges. So for points to be merged, we will say label of higher numbered, higher numbered node will be made equal to the point number or the node number of lower numbered point, so the point number or node number of lower numbered node. That means 28 is to be merged with 20, so label of 28 I will change that to 20. That's what I am saying here. The label of the higher numbered node, the higher numbered node is 28, its label we put equal to node number of the lower numbered node. The lower numbered number is 20, so its label will be changed to 20. So instead of this minus 1, I will changes this to 20.

Similarly between 23 and 19, the label of 23 I will change that to 19 because 23 is to be merged with 19. So label of 23, I will change that to 19, not to the label of 19 I will change it to a number 19. Similarly label of 28 will be made 20. Is the step okay? Once this is done then we come to the main numbering loop.

Now you start assigning node numbers and what we do is if I will write the label of the point is less than 0 then we will say node number is equal to let's say some number NN and we will say NN is equal to NN plus 1, initialize as NN equal to 1. We do this initialization and then we cover all the nodes and we say if the label of the point is less than 0, the node number is equal to NN. That means we will start with node number point number 1, if its label is negative, its node number will be made equal to NN. We have initialized NN as 1, so this will take a node number of 1.

Then I will say NN equal to NN plus 1. I will come to the second point, again the node number is negative and this will become 2. This point will become 3, this will become 4, this will become 5 and so on. So node number of this node will become 1, for this node it will become 2 and so on. So this will become 6, this one will become 7, this will become 8 9 10. Similarly 11, this one will become 12, this will be 13, this will be 14 15 16 17 and for all these nodes right now the label which I have written in blue over here is negative. So the node numbers will remain the same for all of them 19 20. Here also the node number is negative, this will become 21, here also the label is negative, this will become 22 but here the label is not negative. So what do we do in that case where this is only when the label of the point is less than 0.

(Refer Slide Time: 00:28:02 min)



So if the label of the point is positive, we can say if the label of the point is greater than 0 then we will say node number will be equal to node number of point numbered as label, the node number of the point numbered as label. That means for node number, for point number 23, the label is positive. So the node number for 23 will be the node number of

the label that means node number for 23 will be the same as the node number for point number 19 and for point number 19, the node number is 19. So for this point, the node will be given as 19. Student: storing the information also in an array, for any y its label number its node. Yes yes. Again. Why? See, right now it's a very simple thing, so you can make out which two nodes are being merged. But let's say if you have a number of elements and made the other let's say in this case if you have 2 edges being merged here and 2 edges being merged here. We also might have situations were more than 2 edges are being merged together. That means 9 10 has being merged with 11 12 which is also being merged with 7 8. In such case, deciding just by looking at the figure is not going to be so easy. Right now we have taken a very simplistic example and especially when this is extended to 3 dimensions just deciding, giving a numbering directly is not going to be possible. So we have this, it's a simple labeling mechanism.

I am just saying that I am maintaining both the arrays and in this step, this step is just equivalent to saying that the node number of the point will be equal to node number of the label of the point. Student: It means that because falling to which level which node number is there, I mean for the point. I didn't get you. Student: Corresponding to which I mean for a node number we know that which level number it is. I am storing for every node, for every point sorry for every point I am storing the label that label is what I have written here in blue minus 1 minus 1, 0 is here and 19 20 here. I am storing this as the label and finally a node numbers to be given to this point, that I am finding out, I am showing that in an array. That array is let's say an array N, N of point just one sec. N of point will finally store the number I am giving to every point and I am having another array which is the label.

So initially I am making an array called label and then I am using this array to make an array called point. Is that okay? What is the first array? The first array is the label for every point. Initially minus 1 for all the points, for the void blocks whichever points are inside the void block I am putting them to 0 and the edges which are being merged, I then change the label of those points to the point numbers of the lower numbered point. So I get this array as label. Once I have this array as label, I use that array to give a number to every point and that is going to be my global node numbering in my final finite element mesh.

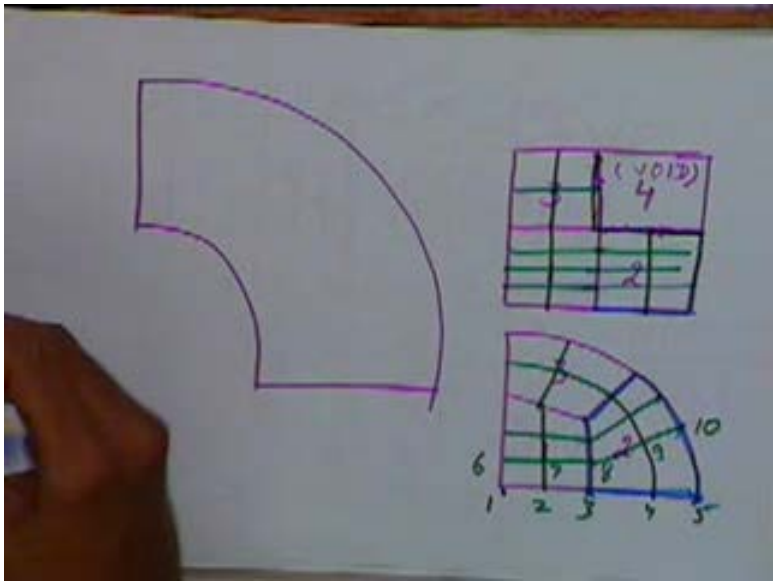
Student: Sir, how do you saying that the node number of 23, the label of 23 is assigned 19. The label of 23 is 19. How do you do that? How do you identify 19? We said that this edge is to be merged with this edge. So, all the nodes on this edge are to be merged with this edge, so 18, 23, 28 is to be merged with 18, 19, 20. so 19 is to be merged with 23, 20 is to be merged with 28. Student: Could again you declare for another object, to identify which edges to be merged. That is the job which has to be done at one of the earlier stages in the algorithm. You decide, identify edges to be merged. You first identify the blocks identify, identify void blocks and identify the edges to be merged and that has to be done at a stage when we are trying to get, generate this shape form a rectangular array.

Actually the basic thing is we are starting with a rectangular array and this rectangular array we want to map this onto any arbitrary shape. So that is an operation which will

require some conceptual mapping that has to be done by the person who is trying to generate the mesh that is not being done in an algorithmic manner. But once that mapping is done then we can have a simple labeling algorithm to generate this mesh. The algorithm is very straight forward. We are just generating the labels and on the basis of that we are getting the node numbers.

Any other question on this? So, if the label of the point is less than 0, we just assigned the node number. If the label of the point is greater than 0 then we assigned the node number corresponding to the node number of the label but if the label of the point is equal to 0, we simply ignore it because if the node number sorry if the label of the point is equal to 0 that means its inside a void block so that should not be numbered. So we will get node numbers corresponding to each of these points. Is that okay? So this is as far as getting the node number is concerned. What we also need to get is the geometric information. That means what are the geometric coordinates for each of these points and how can we get geometric information.

(Refer Slide Time: 36:40)



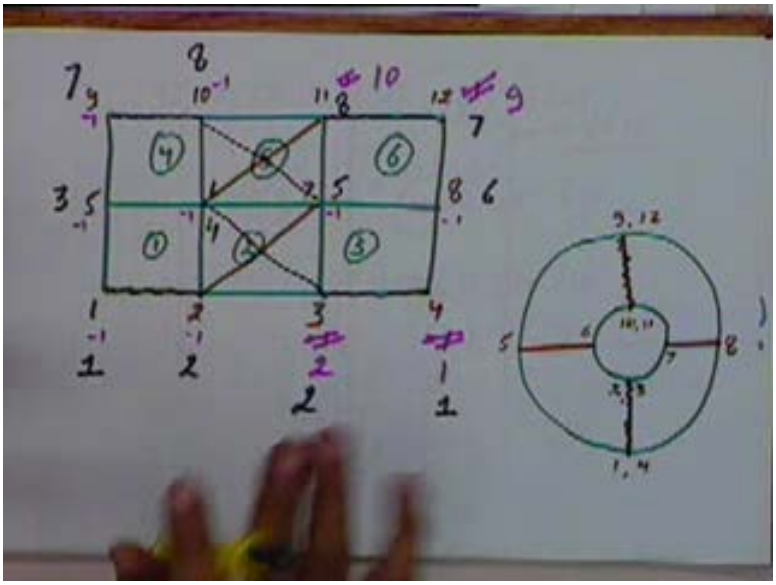
We have started with a shape like this. So if I consider let's say block number 2, in this block number 2 I have, this is one edge and this is the second edge, third and fourth. So this is one edge, second edge, third edge and a fourth edge. I have the geometric information for this object, not this edge list we can take these nodes to corner points and then for getting the intermediate points, these points all these points we can do a simple quadratic interpolation between the nodes. We can get these points by doing a quadratic interpolation.

So effectively the actual element that we will get will have a straight edge over here. Similarly a straight edge here and a straight edge here, if we are taking elements with straight edges. If you have a 4 noded quadrilateral element, it has only straight edges. If you have a 8 noded quadrilateral, it will have a quadratic edge. So if you are taking

element with straight edges, you will get straight edges on all these locations. And these points will be obtained by quadratic interpolation between these two. These two points will come let's say in this case, this edge will correspond to this (Refer Slide Time: 38:08). I know this point and I know this point and I can do a quadratic interpolation between them to get this edge.

So each of these edges, each of these nodes their locations can be obtained by quadratic interpolation and their numbers can be obtained by this algorithm and element numbers will again be given to these, let's say blocks in a row. This is one element let's say e_1, this is e_2 e_3 e_4 and so on. Any question with respect to this mesh generation algorithm? Student: Sir, why don't we for example for assigning the node number for 23, why don't we say node number for 23 is the same. Is assigned its label? Label, because the node number for 19 might not be 19, in this case it is 19. The label for node 19 is minus 1 but the node number for point 19 might not be 19 because right now we have only one edged to be merged.

(Refer Slide Time: 42:02)



In the other example that we had, when this edge is being merged with this and this edge is being merged with this. That is if you do a simple algorithm for this quickly, this will be minus 1 minus 1 minus 1. Initially everything is minus 1, 1 is to be merged with 4 so this will be changed to 1. 3 is to be merged with 2, so this will changed to 2 and here 12 is to be merged with 9, so this will be changed to 9 and this will be changed to 10. Now if we start doing the node numbering, this will take a node number of 1, this will take a node number 2, this will take a node number, will take the label of this and the node number of that, this will become 2. This will again become 1, this will become 3, this will become 4, this will become 5, this will become 6, this will become 7, this will become 8.

Now, 11 that is being merged with 10, so the node number for this should not be 10 but it should be 8. So, we cannot give this the node number, we cannot give this, the node just

as this 10. We have to give it as the node number of this node 10. So this will be, the node number given to this will be 8 and given to this will be 7. I am I haven't taken a finer mesh right now, you can take a finer mesh and repeat the algorithm. This is not the only algorithm for mesh generation. This is one algorithm.

We have other methods available which you are some of you might have tried on pattern that you can give meshes and get the mesh automatically and so on. The other algorithms also which are available but this is one simple algorithm which gives some idea of how meshes can be generated. Any questions on this part? There are no questions. With this I will wind up the finite element portion and in the next class, we will now start with curves, surfaces and solid modeling. So, that's all for today.