Finite Element Method

Prof C. S. Upadhyay

Department of Mechanical Engineering

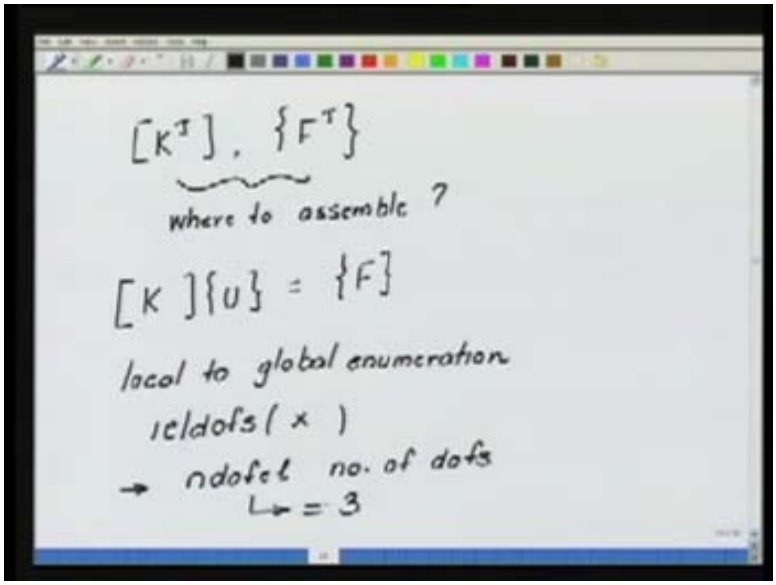
Indian Institute of Technology, Kanpur

Module – 7 Lecture – 4

In the previous lecture we talked about how to compute the element stiffness matrix and the element load vector entries for the linear approximation of the boundary value problem that we had taken. At the same time, while we are doing the element calculations, we can also compute the contribution from the non-homogeneous Norman part of the boundary. That is all that we have to consider as far as contribution of this boundary fluxes or tractions are concerned.

We now have the element stiffness matrix and the load vector as we had constructed in the previous lecture.

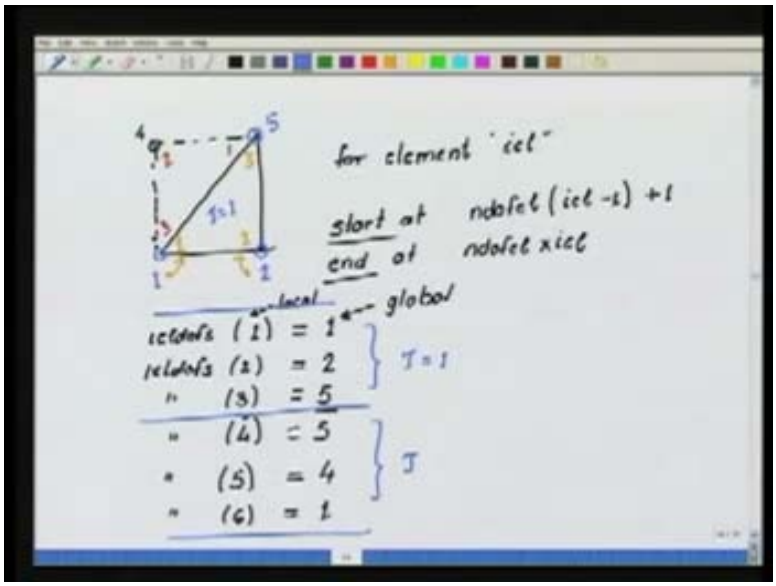
(Refer Slide Time: 01:10)



We have for an element, the stiffness matrix K tau and the load vector F tau. Next, the question that we had asked in the one-D problem and which we will ask is where do we assemble? That is, in the global stiffness matrix K, the global problem [K] {U} = {F}.

Which row and which column should I add the entries of the element stiffness matrix and similarly for the load vector? For this again we have to have a local to global enumeration, for which we had defined for the one-D case, the construction of an ieldofs array or ieldofs vector. We can have it either as an array or as a vector, but here we will choose to have this ieldofs as a vector. In every element we will have ndofel number of degrees of freedoms. For our case of linear approximation, ndofel is equal to 3. Our job is essentially to find the local to global enumeration for 3 degrees of freedom at the element level.

(Refer Slide Time: 03:22)



Let us take again our first element in the domain that we had made in the previous lecture. I will talk of the degrees of freedom as this. This is tau equal to 1, this is the global node 1, global node 2 and if we remember this is the global node 5 and we had said when we do the local enumeration, this is 1, this is 2, and this is 3. Remember, local enumeration is always done in a counter-clock wise manner.

How do I pass this information to my assembly routine? That is, how do I tell that the local 1 corresponds to the global 1? This again, I can do while we are making the mesh, while we are making the nodes of the elements. In this case it is quite simple. We see that as we store the names of the nodes for the element in the nodes array; similarly, I can

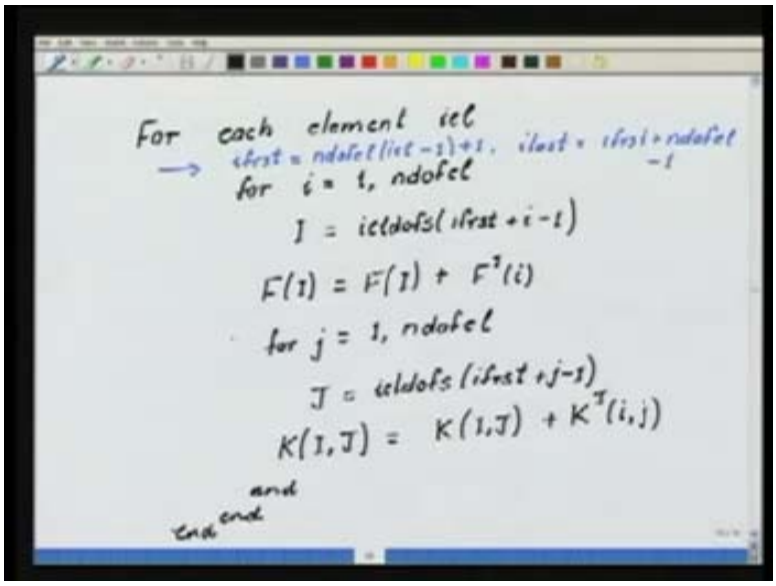
give the same numbers because the global degrees of freedom are also numbered the same way as nodes because they correspond to the nodes now. We can give the same numbers to the ieldofs. How will I do the ieldofs numbering?

ieldofs for an element ‘iel’, the starting point will be: start at ndofel into iel minus one plus one. For the element there are ndofel degrees of freedom. The degrees of freedom for the element correspond to ndofel into iel. These are the locations in the ieldofs array which are going to give me the first degree of freedom corresponding to the element and the last degree of freedom for the element.

For example, if I go by this, then ieldofs one (because for the element one iel minus 1 is 1). The first location will be equal to the local one corresponding to the global number. The local 1 is equal to the global 1. Similarly, ieldofs 2 for the element will be equal to the global 2 and ieldofs 3 for the element will be equal to 5. ieldofs 1 to 3 correspond to the degrees of freedom of the element 1. If I go to element 2, we will have 4 as a degree of freedom and for the element 2 this will be 1, this will be 2 and this will be 3.

For element 2 I will have the starting ieldofs correspond to 2 minus 1 is 1 into 3 plus 1. The ieldof 4 will be the first local degree of freedom for element 2 and this will correspond to the global number 5. ieldof 5 will correspond to the second local degree of freedom of the element 2, that is, 4. ieldof 6 will correspond to the third local degree of freedom of element 2 and is equal to 1. We can imagine that I have these blocks. This is for tau equal to 1 and this block is for tau equal to 2. This way I can arrange the entries of the ieldofs vector and this can exactly tell me: What is the enumeration? What is the correspondence of the local degree of freedom with respect to the global degree of freedom?

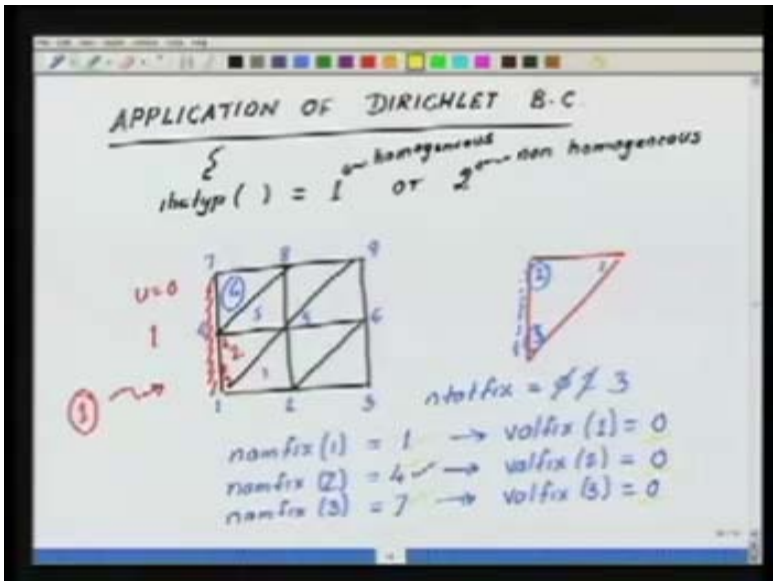
(Refer Slide Time: 08:27)



What do I do using this information? I start my assembly loop for each iel. For each element iel, for i = 1 to ndofel, (this is the local i) I will find the global I. This is equal to ieldofsl. For the element iel, I will find here the ifirst is equal to ndofel into iel minus one plus 1. ilast is equal to ifirst plus ndofel minus one. I find exactly where in the ieldofs array the degrees of freedom corresponding to the element of setting is. Then I will come here and I will be equal to ifirst plus i minus one. We see that i = 1 will correspond to the ieldof setting in ifirst plus i minus one. Then I will do F, I will now assemble the global F. Remember that before doing assembly these F and K matrices should be initialized to zero. F(I) is equal to F(I) plus F for the tau i.

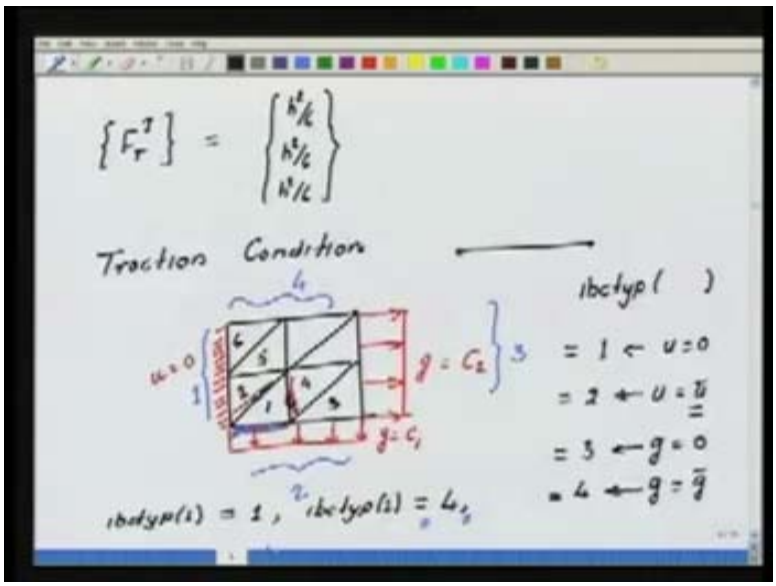
Embedded loop for j is equal to one comma ndofel, J is equal to ieldofs ifirst plus j minus one. This gives me the global name of the jth, the column degree of freedom and then I will have K I comma J is equal to K I comma J plus K tau i comma j. As far as the assembly is concerned, we are now able to assemble the stiffness matrix and load vector corresponding to the element. I now end the loop over the rows, columns and over the elements. This is all I need to do as far as the assembly is concerned. What is left now?

(Refer Slide Time: 12:05)



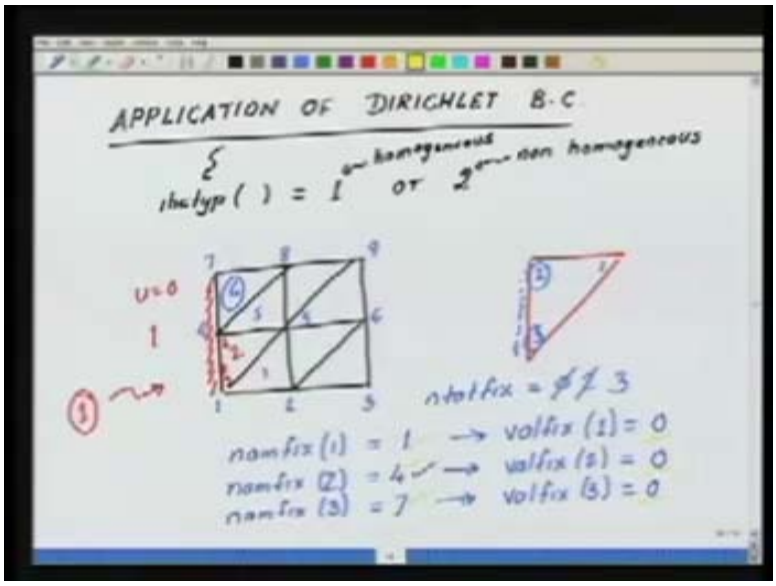
What is left is application of dirichlet boundary conditions. Here this corresponds to ibc type equal to one, which is the homogeneous dirichlet. Let us see our numbering.

(Refer Slide Time: 12:48)



We will have the one and the type two which will correspond to our element.

(Refer Slide Time: 17:35)



Here I will have either one – the homogeneous dirichlet or two, which is the non-homogenous dirichlet, both of which have to be enforced in the global stiffness matrix and the load vector. How do I find out what are the degrees of freedom that have to be fixed? Again, I will draw the simple domain that I had taken. I will write the global nodes and the degrees of freedom and I will remember that this was edge 1 on which I had ibc type two. That is, this is the only edge where the dirichlet boundary conditions had to be applied because u was equal to zero on this edge.

The question is that if I have to apply the dirichlet boundary conditions on this edge, that is, I have to fix the displacement u along this edge to be equal to zero and all the degrees of freedom setting on this edge have to be made equal to zero, then how do I identify those degrees of freedoms? When we are finding the neighbors of the sides, when I go to element two for example, I find that the second edge of the element is sitting on the boundary segment one. I immediately know that for the boundary segment one the boundary condition is of a homogeneous dirichlet type. It is not a non-homogeneous it should be one. Boundary type should be one. That is, ibc type should be one.

As soon as I do that, I find that this is an edge on which the dirichlet boundary conditions have to be applied. As I loop over the elements to find the neighbors after making the

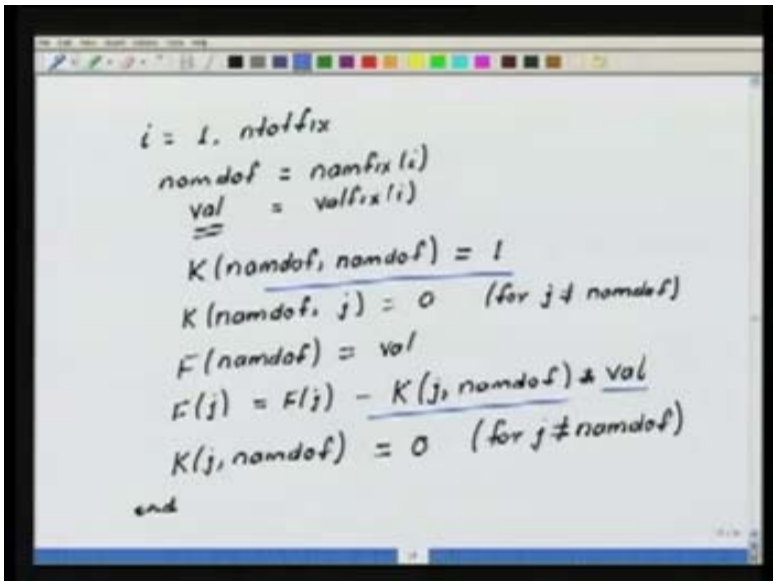
mesh, then immediately I can also make a list of the degrees of freedom that are lying on the boundary. To make the list of the degrees of freedom lying on the boundary, I go to the particular edge, which is the second edge of the element. Which are the degrees of freedom, which are non-zero and are active on this edge? I find that for this edge the local degree of freedom are two and the local three. If I draw the second element, this is the local one, local two and local three. These are the ones which are lying on this edge which have to be fixed. Immediately, I start a counter and start counting the number of degrees of freedom, which have to be set to zero.

I will put start total number of fixed degrees of freedom is equal to zero initialize. Now I loop over these edges and I have name of the fixed node. I increase the counter. For example, for first one will be equal to the global node one. Name of the second node that is fixed from this element comes out to be the global node four. Then I go to the other elements and as I am doing this assigning of the degrees of freedom and assigning of the neighbors on the edges, I come to element six. When I come to element six, I find that element six is sitting on boundary segment one, which is the boundary type one. Which means that here also the displacement conditions have to be satisfied.

I come to this one. So my counter would have become two at the end assembly of element two. When I come to element six, I check whether a degree of freedom that lies on a fixed edge of this element six is already accounted for or not. I find that this element six has again its second edge on the boundary and that has global degrees of freedom four and seven as the fixed degrees of freedom. But then I find that four is already accounted for. If it is accounted for, I do not add it again. But then, I find seven is a new degree of freedom. Therefore, this is equal to seven. This counter has become three. I know that there are three degrees of freedom coming out of the dirichlet boundary conditions that have to fixed. The names of these degrees of freedom are this and while I am giving the naming, at the same time I look at the type. I can find the value of the fixed degree of freedom. We need not have only zero boundary conditions. We can have a fixed displacement or a fixed temperature that is non-zero on the edge.

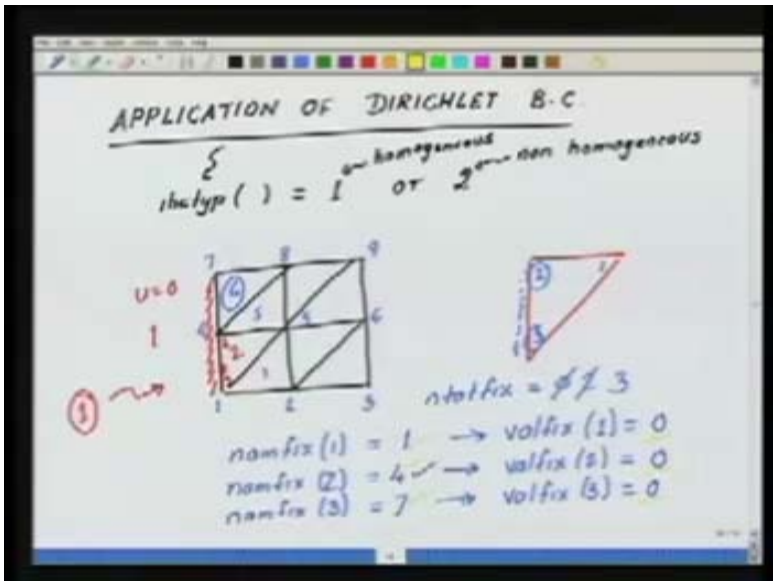
I will say that the value of the first fix degrees of freedom comes from the information given for this boundary edge as far as the temperature is given. In this case it is zero. Similarly, I will have the value of the fixed degree of freedom for the second degree of the freedom that is also zero; for the third degree of freedom also it is zero. I know exactly which degrees of freedom have to be fixed by this numbering here and I know exactly what is the value that has to be given to it. All these information is available to us. Now what do I do?

Refer Slide Time: 20:13)



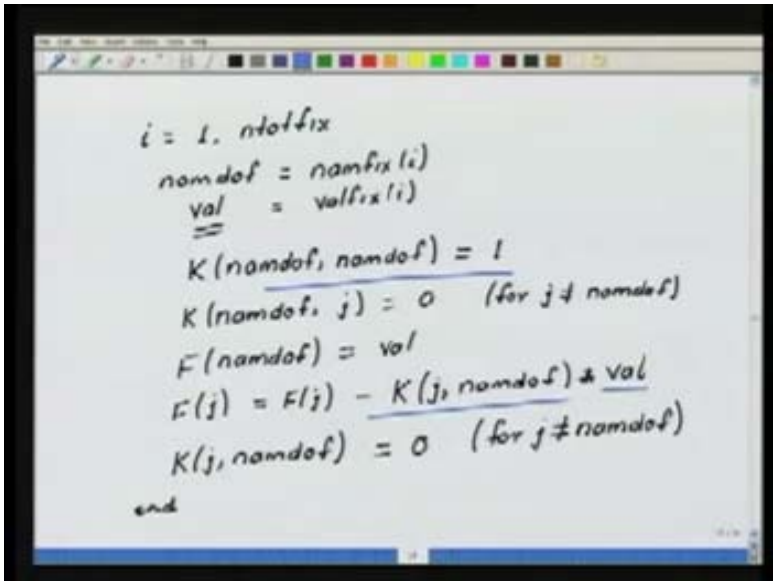
I go to my global stiffness matrix and I start loop over i is equal to one to ntotfix, that is, the total number of fixed degrees of freedom. Find name of the degrees of freedom is equal to what we have given in the previous slide.

(Refer Slide Time: 20:32)



It is namfix i, val is equal to valfix i.

(Refer Slide Time: 20:37)



I know which global row I have to go to and I know what the value is. After that I will simply say that K (namdof, namdof) is equal to one. Then K (namdof, j) is equal to zero, for j not equal to namdof. What have I done? I have gone to this particular row which corresponds to the given global degree of freedom and I have made the diagonal entry of

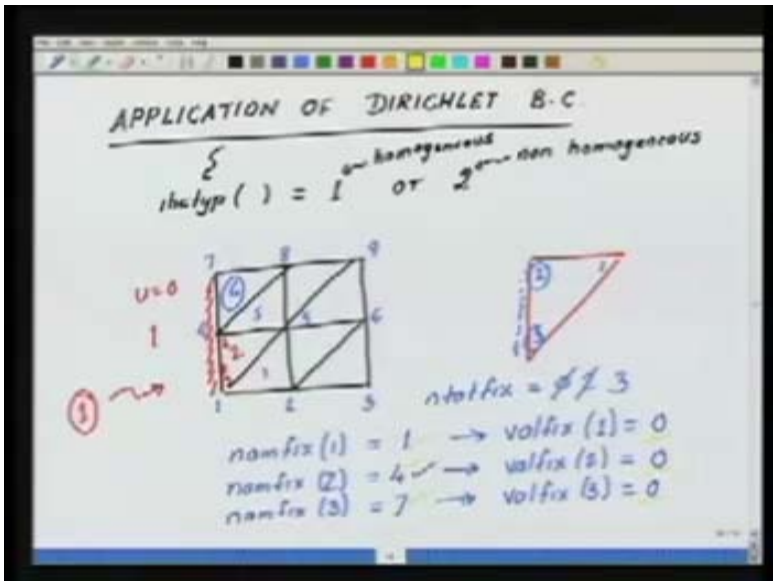
that row equal to one and all the off diagonal entries of that particular row equal to zero. Once I have done that and this row is taken care of, then F namdof is given the value that is the fixed value. Therefore, it is val.

After this, what do I have to do? I have to also correct the remaining equations. In the remaining equations F(j) is equal to F(j) minus K(j, namdof) into val. I am going to the other equations and accounting for the known value of the given degree of freedom by taking the multiple of that corresponding column entry against the known value of the degree of freedom to the right hand side, because this is a totally known quantity which modifies my load vector and then after that K(j, namdof) is equal to zero.

Once I have done this, I end the loop. Once I have done this, I have taken care of the degree of freedom in its corresponding row fixing it by modifying the stiffness entries in such a way that it gives me the desired value exactly the same way as we had done in the one-D case and we first modified the load vector in the other rows to account for the known value of the degree of freedom, in that we are trying to eliminate from the system and then set the corresponding row entry to zero.

Once I have done this, the dirichlet boundary conditions are also taken care of. We see that when the val is equal to zero, that is, I am talking of a homogeneous dirichlet condition, this quantity leads to no correction in the right hand side. But we have written it in a generic way so that this can be used for all types of homogeneous and non-homogeneous dirichlet boundary conditions.

(Refer Slide Time: 24:52)



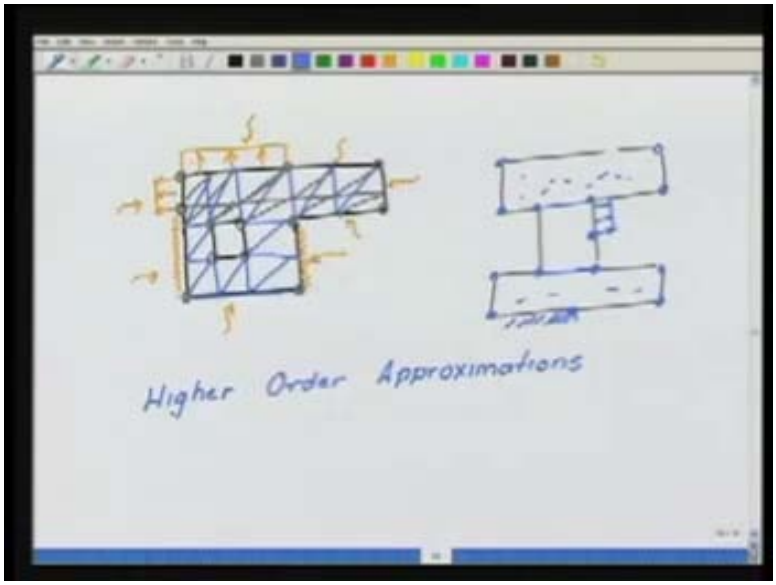
One thing I like to point out again is that we are applying the boundary conditions on the edge and not only at the nodes. There is always a misconception that the dirichlet conditions have to be applied at the nodes and that the force conditions have to be also applied at the node. We are doing integrals along lines, along edges and those integrals actually translate to fixing of values or assigning of loads to the degrees of freedom sitting on the particular edge. This is something that we have to always keep in mind.

We should be able to solve the problem. If I look at the fifth equation, that is, the equation corresponding to the global five degree of freedom, this equation will have contribution from elements one, two, four, five, seven and eight. It has contributions from six elements, because, there are six elements which are sharing this node amongst themselves.

Similarly, if I am talking of the third node, this will have only contribution from the third element. No other elements actually participate in this node. We see that I can have contribution to a node coming from one element, from two elements, from five elements, from a varying number of elements. It is not the same as what we had in the one-D case where at most two elements participated in the equations for a node. Here we could have, depending on the mesh, many more elements present.

Another thing is, if I look at degree of freedom one, it actually participates in the equations in the definition of u along the boundary edge two as well as along the boundary edge one. When we say that we are going to fix the u on an edge, it forces us to fix the value of the degree of freedom one. In this case, the value of degree of freedom one is actually equal to zero. The dirichlet condition actually predominates over the Norman condition. That is, the value of the degree of freedom, if it is fixed from one side, then that is what we are going to assign to that particular degree of freedom. And we should not confuse that from one side we get a load contribution and from the other we get a displacement contribution because, these things are defined only on the edges. With all this done, we should now be able to assemble the global stiffness matrices, the global load vectors, apply the dirichlet boundary conditions and solve the problem. That should not be a problem as far as the simple problem that I have taken. Let me ask another question.

(Refer Slide Time: 28:12)



I have a domain like this; I could also have an internal cutout. First, let us make a simple mesh over this. If there is a cutout here, I am making a very crude mesh. Our first job is always to make the mesh. Before making the mesh, what have I done implicitly? I have identified the geometrical entity. This domain has these distinct edges: edge one, edge two, edge three, edge four, edge five, edge six and the four internal edges which

correspond to the boundary of the whole. It has actually ten such edges. I have not done anything as far as the loads are concerned. Let us say that here I am fixing the specimen. Here I am applying traction. Here in this region also I am applying traction and I may be fixing it in this region.

As far as the definition of the geometry is concerned, we see that I need only definition of these ten edges but as far as the boundary condition is concerned, this edge has one boundary conditions specified which is a zero flux, this has another boundary condition applied, this edge has another condition, this one has another, this one has another. We see this edge has a different boundary condition, while this edge has a different one and this edge has a different one. All these edges also have their specific boundary conditions applied. A global geometric edge may actually have sub-sections where boundary conditions are changing and we have to honor this transition or change of boundary conditions by putting nodes at those points where the boundary condition transitions.

This mesh is actually unacceptable because here is a point where the boundary condition transitions and I have not put a node there. In a very simplistic way I should have a line like this. I am just simply dividing the mesh and if I have this line then I will have to mesh it like this. There are times when in fact this is always going to be the case in practical problems that the boundary conditions decide how many boundary segments we are going to define a priori and from there we have to first honor the existence of this transition boundary conditions while putting nodes. This has to be a node, this has to be a node, this has to be a node, and this has to be a node. All these have to be nodes because the boundary conditions are transitioning, as well as, in certain points, the geometry is transitioning. For example, here the boundary conditions are similar, but the geometry has a change. I have to honor those. These nodes I have to put. So these are the minimum number of nodes that I have to put in the definition of our mesh.

The new mesh is no longer what I have drawn earlier it is actually these entities. In most problems, one has to be careful that we do not violate the applied boundary conditions. Another thing I may have is I have different material domains. Nobody tells me that my two dimensional domain will have the same material everywhere. I could have absolutely

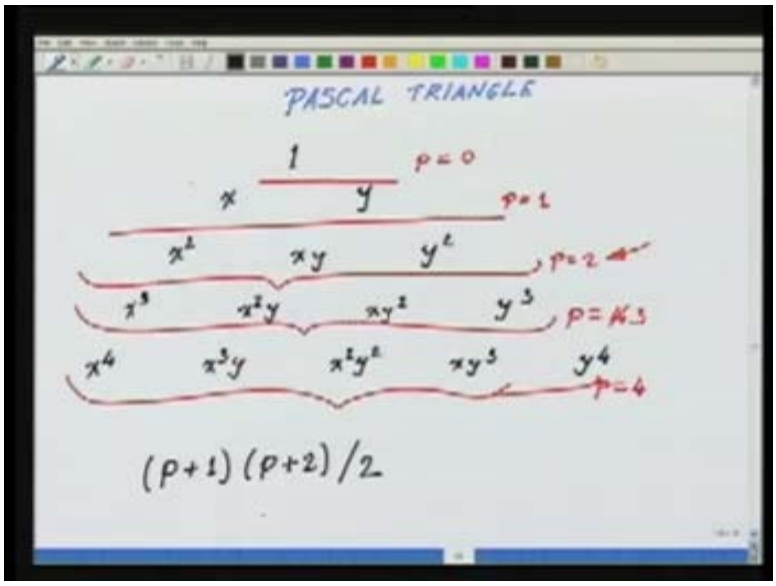
different materials in different regions. For example, I take this sample. I decide to put one material here, one material here and something else here.

These material boundaries also have to be clearly honored when we are talking of making a mesh. These material boundaries have to be demarcated and we also have the other ones corresponding to geometry. And then, depending on the boundary condition for example, here I may have traction applied for a flux condition. This point also has to be a node. I may fix this part, so this point will also have a node here. We will have to have these nodes put apriori, as soon as the boundary information, material information and the geometry information is available to us. Then we start meshing. The rest of the mesh can come after this. These things have to be kept in mind.

We have done enough with these linear approximations. We have gone through the whole process of what we will be actually doing in a finite element setting. The only thing which is missing in what we have done is that the integrations have been done in an exact way. We would like to add numerical integration, which we will talk about later. I may have to do better than linear approximation to get a reasonably accurate solution.

If I have to do better than linear approximation, I should be able to make higher order approximations. When we are talking of higher order approximations, we will be talking of piece-wise polynomials. The issue of completeness and the issue of linear independence will all come up. As far as two-D is concerned, the higher order polynomials are given, to which, all the monomials that have to appear in a given order polynomial for that representation to be complete, is governed by something called a Pascal triangle.

(Refer Slide Time: 35:58)



If I take a monomial corresponding to the zeroth hour polynomial, it is one. If I take a monomial corresponding to the linears, it has x and it has y. If I take this correspond one x y as a monomials corresponding to linear. If I take quadratic, I will have x squared, xy, y squared. If I take cubic, I will have x power three, I will have x squared y, I will have x y squared and I will have y power three. If I am talking of fourth order, then I have x power four, x power three y, x squared y squared, x y power three and y power four. Constant monomial, order zero polynomial will require only one monomial in its definitions. We will call the order P equal to zero.

Linear approximation or the linear representation will require three monomials: one x, y. A quadratic representation will require six monomials (all the things that lie above are also included): one xy, x, xy, y squared. A cubic will require ten monomials. That is, to all that which was there for the quadratic, I add x power three, x squared y, x y squared, y power three, that is ten. And the fourth order will require fifteen monomials in its definition.

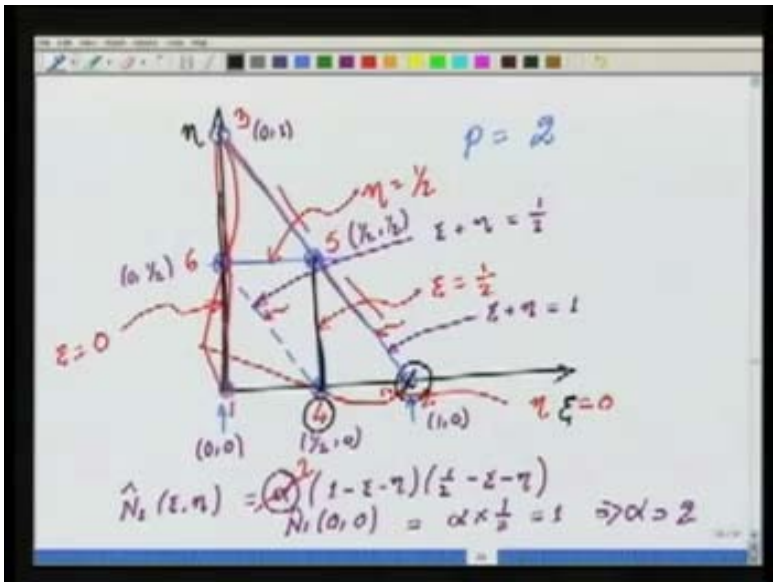
When we talk of construction of basis, when we talked of the linear, we saw that we had three and the basis was linearly independent and complete. When we go to the quadratic approximation over a triangle, we actually need to have six basis functions defined in the

element. That is, six independent functions of degree two have to be defined in order to completely represent a quadratic polynomial in the element.

How are we going to do these things? We see cubic will require the ten. The generic definition is that the number of monomials corresponding to a given order of representation will be equal to the number of monomials given by P plus one, P plus two, divided by two. We will see that when P is equal to one, this is two into three by two, which is equal to three. When P is equal to two, this is three into four by two, which is equal to six and so on. We will get out of this, exactly the number of monomials on independent basis that have to be defined in order to completely represent the given order polynomial.

This has to be used in a definition of or in a construction of these higher order basis functions. I am going to operate in the master element only.

(Refer Slide Time: 39:52)



Let us take the generic master element. This is psi and this is eta. I am going to make the element like this. We have made the generic master element tau hat. As far as the linear approximation was concerned, I had the basis functions defined with respect to these three nodes. When I want to do quadratic, I will borrow a lot of ideas from what we have

done in two-D. If I am taking a quadratic approximation in this triangle, then the projection of the quadratic say on the first edge should also be quadratic because it is a quadratic in a triangle. If the projection of the quadratic on the first edge is a quadratic and I have the basis functions on this edge defined with respect to the corners, then the simplest way I can define a quadratic on the edge is put a middle node here.

Similarly, I look at this edge. When I look at the projection on this edge, (I know that the quadratic is defined over the whole triangle), it also has to be a quadratic. So why not define a middle node here? We draw the line parallel to the base through this middle node and the line parallel to the base through this middle node will cut at the middle of this edge. And as we have said for these two edges, similarly on this edge also, the projection of the quadratic should be a quadratic, so this edge also requires a third edge node to be added. We see that in order to have the quadratic at least on these edges, we have to have these three new nodes. Then I will complete this element. We have one, two, three, four, five and six nodes. The question is, can six be the number of independent basis functions that we require to define a quadratic approximation?

We have six nodes. Now can we define a basis function or a shape function in the element such that the six shape functions correspond to a value one at their corresponding node and zero at all other nodes and they form a complete basis for the quadratic? Our challenge is: let us say, I want to define $N_{1.}$ So I call this node one. I will do the numbering in a way which is a little different from the standard 1, 2, 3, 4, 5 and 6. If I want to define N_1, N_1 will be such that it is one at this point, becomes zero at this point and zero here. Similarly, on this edge it is one at this point, zero here and zero here. N_1 is a curved roof that gradually dies off too value zero on this second edge of the master element.

How do I now define a quadratic that is one at this point and zero along all these points and at these points? It is quite easy. We see that if I take a function that vanishes along this line and multiply these two functions, I should be able to get a quadratic. What is the equation of this line? The equation of this line is, if I take the equation of this particular line, it will give me a psi plus eta is equal to half because this corresponds to the point

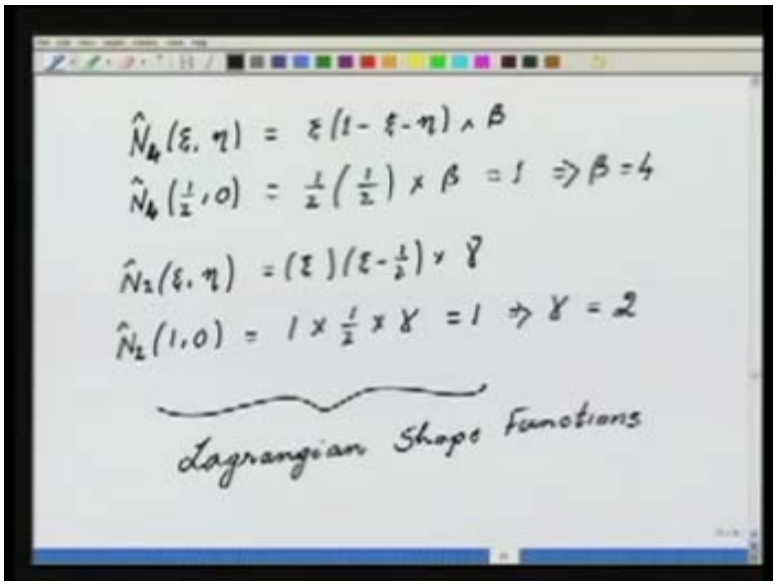
half, zero. This corresponds to the point zero, zero, this corresponds to the point one, zero, this corresponds to the point one by two, one by two, this corresponds to zero, one, this corresponds to zero, one by two. The equation of this line is psi plus eta is equal to half. Equation of this line is psi plus eta is equal to one. If we see that N_1 hat is a function of psi and eta is equal to some factor alpha into one minus psi minus eta into half minus psi minus eta and if I multiply these things out, it is going to be a quadratic and in psi and eta and it is also going to vanish along this line and this line. If it vanishes along this line and this line, it is also vanishing at all these nodes. This is a very simple way of defining N_1 hat. Alpha will come by putting the value of psi eta equal to zero because I want the N_1 hat at zero zero equal to a value one.

What will I have? I have psi eta is equal to zero. So N_1 hat at zero zero is equal to alpha into half, which implies it is equal to one, which implies alpha is equal to two. I can replace alpha with two. That is my definition of a quadratic shape function corresponding to the node one.

Similarly, if I want to define the quadratic shape function with respect to node two, such that this quadratic vanishes at all other points. We see that all other points lie on this line and they lie on this line.

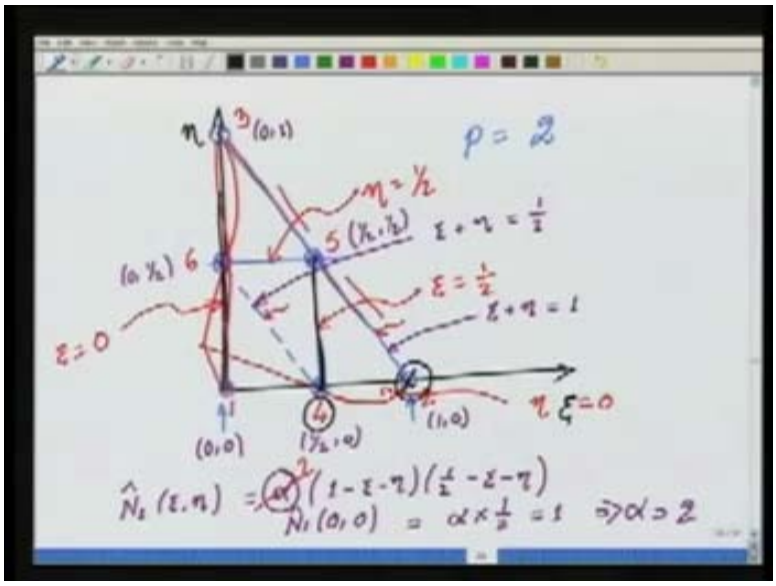
If I take the equations of these two lines and multiply them with the factor in front, that should give me the representation of the second shape function. Equation of this line is: psi is equal to zero. Let me go ahead. The bottom line is that all these functions can be constructed in terms of products of the equations of one of these lines that I have drawn. Equation of this line is: eta is equal to zero; equation of this line is: eta is equal to half and equation of this line is psi is equal to half.

(Refer Slide Time: 48:41)



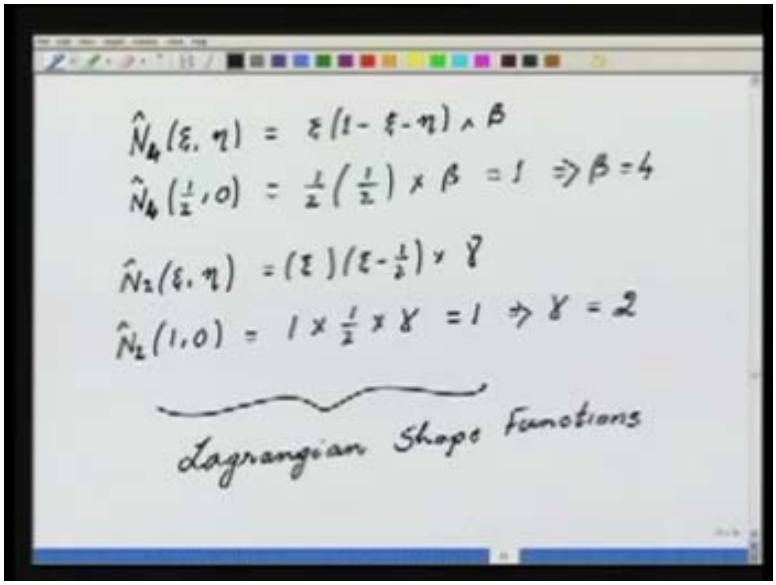
As far as N_2 is concerned, I will take psi into one minus psi minus eta into alpha. We see that N_2 as a function of psi and eta is equal to psi into one minus psi minus eta into some constant beta. Where should N_2 be if we check that this thing will vanish on all the other points? At the point psi equal to half, eta equal to zero, this is equal to half into one minus psi minus eta which is half, into beta, this has to be equal to one implies beta is equal to four.

(Refer Slide Time: 49:26)



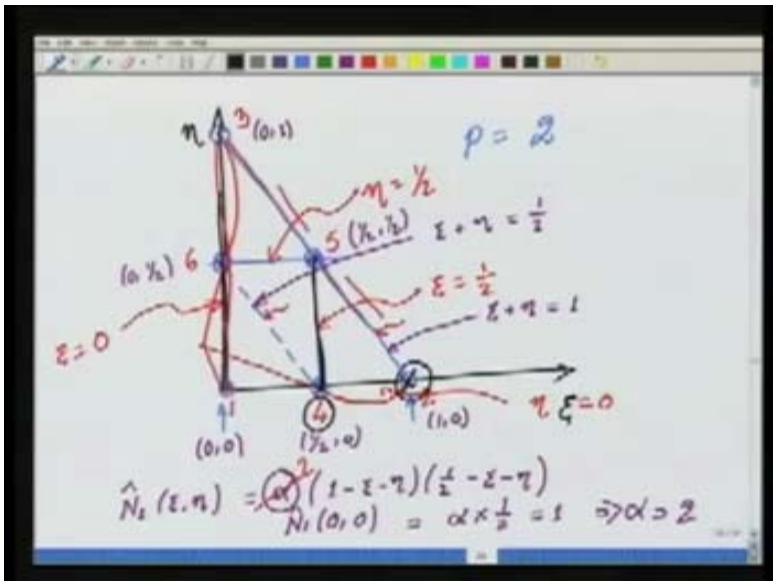
Similarly, this is not N_2 this is N_4.

(Refer Slide Time: 49:30)



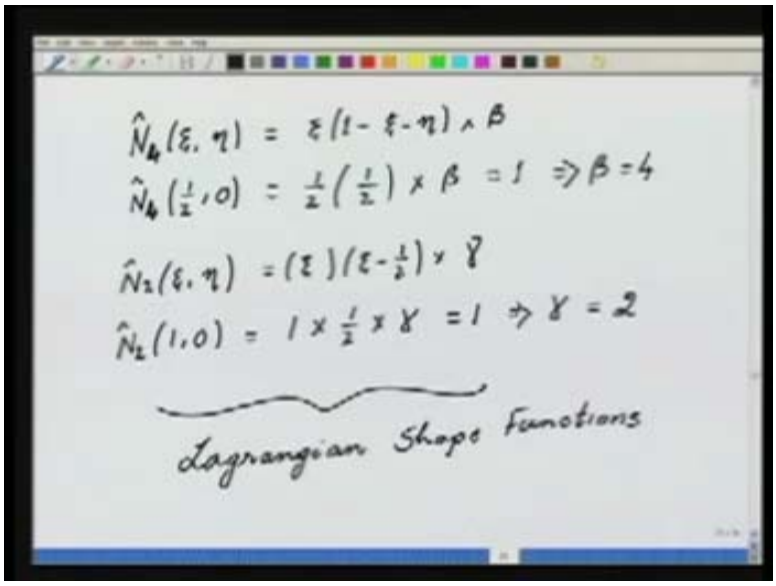
I should go back and change this to N_4.

(Refer Slide Time: 49:35)



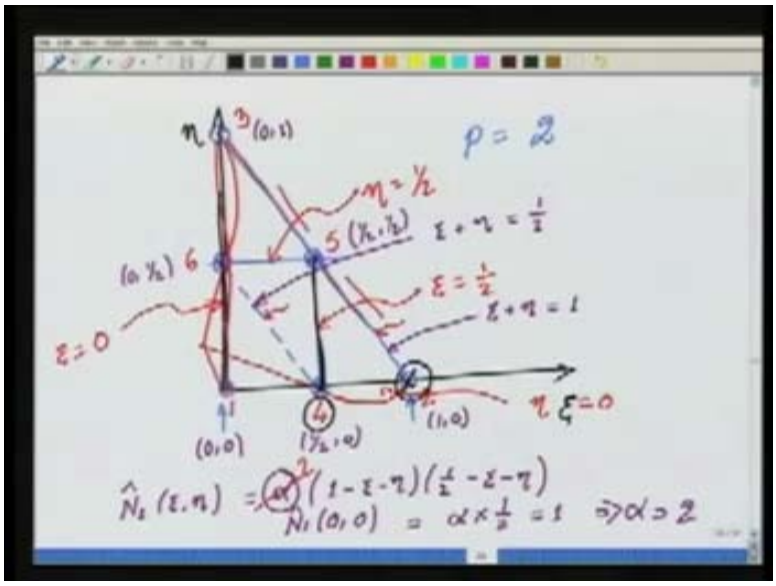
I want to have representation of N_2. N_2 is one which should be having a value one at this node and zero at all other nodes. If it has to be zero at all other nodes I can make that happen by taking this function to vanish on this edge and this edge. What is the equation of this edge: equation of this edge is psi is equal to half and equation of this edge is psi equal to zero.

(Refer Slide Time: 50:05)



I can define N_2 hat has a function of psi and eta which is equal to psi minus zero into psi minus half into gamma. This thing should have a value one at the point psi equal to one eta equal to zero. This is one into half into gamma, which is equal to one, which implies gamma is equal to two. This way I can construct all the shape functions corresponding to the quadratic. Let me take this further.

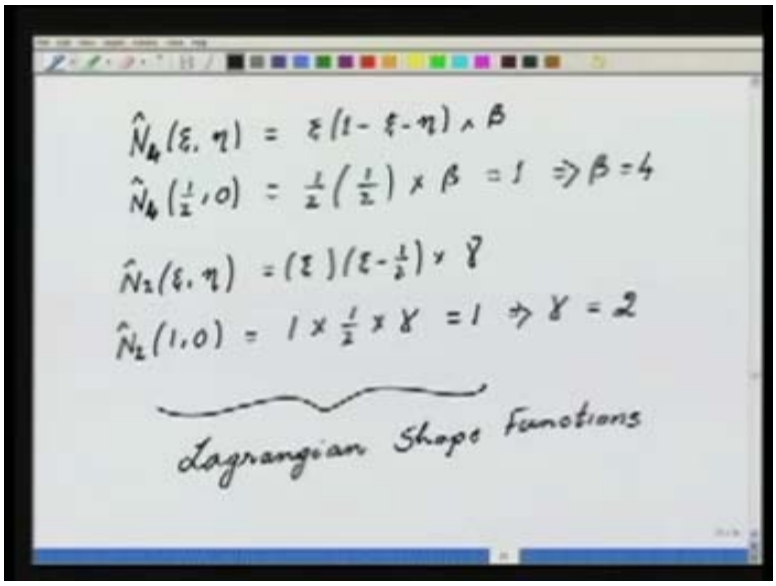
(Refer Slide Time: 50:50)



How will I do N_5? I have N_1 and I have found N_4 and N_2. Let us do N_3. N_3 is one which vanishes if I see along this edge and this edge. N_3 will have equation eta minus half into eta minus zero into some constant. That constant I can find by putting the value of N_3 equal to one at the node three.

Similarly, N_5 is one, which is one here and zero at all other nodes. Which means, zero along this line and zero along this line. I take the equation of this line, which is psi equal to zero and this line, which is eta equal to zero, this one is psi eta and so on. N_6 is very easy. N_6 six will be one here, zero along this line and this line. It is one minus psi minus eta into eta into some constant that is equal to N_6.

(Refer Slide Time: 51:56)



This way I can construct all the six quadratic shape functions. And what I have done is essentially an extension of the Lagrangian definition to 2 D. In the next lecture, I am going to extend this further to the cubic approximation, to the fourth order approximation and then we will go and look at some more families of shape functions that can be used in our practical problems.